TECHNISCHE UNIVERSITÄT
CHEMNITZ

Fakultät für Informatik

Professur Rechnernetze
und verteilte Systeme

Diplomarbeit

Wireless-LAN im Studentennetzwerk (CSN)

Autor: Alexander Glöckner

Betreuer: Prof. Dr. Uwe Hübner
Dr. Jörg Anders
Dr. Robert Baumgartl

Datum: 14. Dezember 2005

Inhaltsverzeichnis

1 Aufgabenstellung

Wireless LAN im Studentennetzwerk (CSN)

Das studentische Netzwerk soll um ein WLAN erweitert werden. Es soll eine möglichst benutzerfreundliche und sichere Nutzung des WLAN-Netzes durch alle angemeldeten Nutzer des Studentennetzes möglich sein. Untersucht werden sollen dabei mögliche Authentifizierungsmethoden und mögliche Verschlüsselung/Integritätssicherung des Datenverkehrs.

Verschlüsselungen: WEP, WPA, WPA2(RSN), VPN

Authentifizierung: 802.1x - Radius, Diameter u.s.w.

Im Rahmen der Arbeit soll eine Möglichkeit der Umsetzung und Intergration des WLANs in das bestehende Studentennetz und dessen Nutzerverwaltungssystem erarbeitet werden.

2 Hintergrund

In den letzten Jahren hat die Verbreitung von mobilen Geräten (Notebook, PDA usw.) massiv zugenommen. Der Wunsch, diese Geräte an beliebiger Stelle zu nutzen und mit dem Internet zu verbinden, ist nahe liegend.

Auch den Studierenden in Chemnitzer Studentenwohnheimen soll die Möglichkeit geboten werden, einfach und komfortabel diese Mobilität zu genießen.

2.1 CSN - Chemnitzer StudentenNetz

Das CSN (Chemnitzer StudentenNetz) ist eine Initiative von Studenten für Studenten, um die privaten Rechner der Studenten an das Campusnetz der TU Chemnitz anzuschließen.

Derzeit hat das CSN etwa 1700 Nutzer an das Campusnetz angeschlossen. Der Anschluss ist in allen vermieteten Zimmern der Wohnheime möglich. Damit ist das CSN auch eines der größten Studentennetze. Aufgrund der positiven Kontakte zum Studentenwerk und zum Rechenzentrum wurden diese Dimensionen erst ermöglicht.

2.1.1 Netzwerkanschluss

Das Studentenwerk Chemnitz-Zwickau hat am Standort Chemnitz zehn Wohnheime. In acht von zehn Wohnheimen wurde bereits ein vollgeswitchtes TP (Twisted Pair)-Netzwerk mit 10 bzw. 100 MBit/s aufgebaut. Die restlichen zwei Wohnheime besitzen noch ein BNC (Bayonet Nut Connector)-Netzwerk.

Jedem Mieter eines Wohnheimzimmers wird die Möglichkeit geboten, sich am Netz des CSN anzuschließen. Voraussetzung hierfür ist ein gültiger Account beim URZ (Universitätsrechenzentrum) [1] sowie eine bestandene ZIN (Zertifikat Internet-Nutzung)-Prüfung [2].

2.1.2 Netzwerk-Struktur

Abbildung 2.1 zeigt die aktuelle Netzwerkstruktur des CSN (Stand Juli 2005).

Eine ausführliche Erklärung der CSN-Netzstruktur wurde von Markus Schade unter dem Titel „*Netzentwicklung im CSN*" [3] veröffentlicht. Diese Veröffentlichung beschreibt den Netzaufbau vom Stand 2004. Im Jahr 2005 sind einige der im Dokument beschriebenen Planungen in die Tat umgesetzt worden.

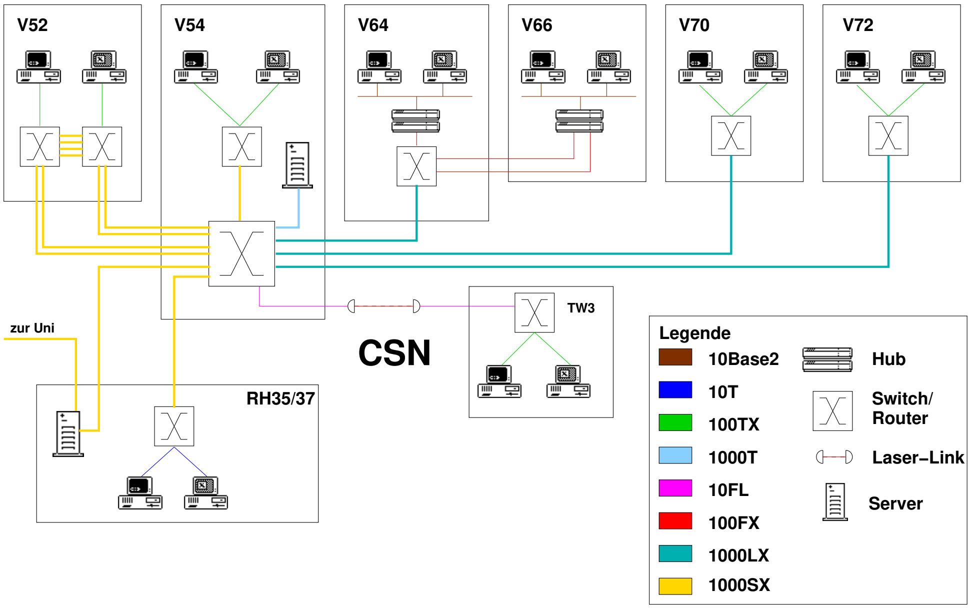

Abbildung 2.1: Netzstruktur des CSN (Stand 2005)

2.1.3 Verwaltungssystem

Im CSN gibt es über 100 Netzkomponenten mit weit über 2400 Anschlüssen für momentan etwa 1700 Nutzer. Um dies verwalten zu können, bedarf es einer großen Anzahl an Automatismen. Das Herzstück dieses Verwaltungssystems bildet ein Datenbanksystem. Hier werden in über 200 Relationen auf 120 Tabellen alle notwendigen Informationen abgebildet.

Anhand dieser Datenbasis werden durch selbstgeschriebene Programme u. a. alle Netzkomponenten einmal pro Stunde konfiguriert. Weiterhin wird damit die Firewall für jede gültige, vergebene IP (Internet Protocol)-Adresse bearbeitet sowie für jeden Nutzer anhand seines verbrauchten Datenvolumens eine entsprechende Bandbreite mittels Shapingsystem (siehe Abschnitt 2.1.4) eingestellt. Außerdem werden stündlich alle Netzdienste (DNS, DHCP) an den aktuellen Datenbankzustand angepasst.

Für die Konfiguration sind verschiedene nutzerbasierende Daten gespeichert, so z. B. die Hardware-Adresse der Netzwerkkarte (MAC-Adresse), die zugewiesene IP-Adresse, der Rechnername sowie der Rechnerstandort (Wohnheim und Zimmernummer).

2.1.4 Traffic-Accounting

Das Datenvolumen von außerhalb des Datennetzes der TU Chemnitz ist für das CSN limitiert. So darf eine maximale Datenmenge im Monat nicht überschritten werden. Aus diesem Grund muss das entstandene Datenvolumen jeder IP-Adresse erfasst und aufgezeichnet werden. Das ermittelte Datenvolumen pro Nutzer wird als Grundlage für das dynamische Shapingsystem genutzt, um damit pro Nutzer eine bestimmte Bandbreite bereitzustellen. Dieses dynamische, nutzerbasierende Shapingsystem wurde im Rahmen einer Diplomarbeit [4] von Jan Horbach entworfen und seitdem ständig weiterentwickelt.

2.2 Einsatz und Nutzung des WLANs

Das Netzwerk des CSN soll durch ein WLAN (Wireless Local Area Network) erweitert werden. Diese Erweiterung des Netzwerkes soll keiner extra Anmeldung für die Nutzung bedürfen. Jeder bereits im CSN registrierte Nutzer soll dies nutzen können.

Als Einsatzorte für dieses WLAN bieten sich die Außengelände der Wohnheime, deren Laubengänge oder die Studentenclubs in den Wohnheimen an.

Momentan ist es nur möglich, den Rechner in dem Zimmer anzuschließen, in welchem er angemeldet ist. Durch die zu schaffende Erweiterung des Netzwerkes soll eine Nutzung des CSN-Anschlusses unabhängig vom aktuellen Standort ermöglicht werden.

2.3 Problemstellungen

Unter den gegebenen Gesichtspunkten muss ein System entwickelt werden, das den oben genannten Anforderungen genügt. Es muss einer großen Anzahl an möglichen Nutzern, die einer großen Fluktuation unterliegt, Zugang gewähren und dabei die Nutzung durch unberechtigte Dritte verhindern. Weiterhin gibt es durch die fehlende explizite Anmeldung des PC-Systems für die Nutzung dieses Zusatzdienstes keine Zugangsbeschränkung auf Basis der Hardware-Adresse der WLAN-Netzwerkkarte. Der Vorteil hierbei ist, dass der Nutzer die Hardware (Notebook, PDA, WLAN-fähiges Handy usw.) frei wählen und wechseln kann. Gleichzeitig ergibt sich der Nachteil der fehlenden Identifizierbarkeit durch die Hardware-Adresse des Gerätes.

Neben der Zugangsbeschränkung sind Datenschutz und Datenintegrität durch entsprechende Verschlüsselungen gegen Spionage und Manipulation bestmöglich zu sichern. Letztendlich sind die Abrechnung der bei der Nutzung des Dienstes entstandene Datenmenge sowie der Netzwerkschutz durch das Firewallsystem, wichtige Faktoren, die durch ein dynamisches Steuerungssystem geregelt werden müssen.

Dieses dynamische Verfahren soll im Rahmen dieser Diplomarbeit untersucht und implementiert werden. In den folgenden Kapiteln werden die zugrunde liegenden Technologien und Standards untersucht, die für die Realisierung in Betracht kommen.

3 Wireless-LAN

3.1 Allgemein

Unter WLAN versteht man ein drahtloses lokales Netzwerk. Hierbei ist in den meisten Fällen ein Standard der IEEE 802.11 Familie gemeint. Mit der physikalischen Schicht beschäftigen sich momentan fünf Protokolle: 802.11, 802.11b, 802.11a, 802.11g und 802.11n. Vier dieser fünf Protokolle sind dabei auf dem freien Markt erhältlich. 802.11n soll noch im Jahr 2005 verabschiedet werden.

Ursprünglich war im IEEE 802.11 Standard bereits das Thema Sicherheit enthalten. Später wurde dies jedoch Teil anderer Standards. Für Serviceverbesserungen, Serviceverlängerungen bzw. Korrekturen zu den vorangegangenen Spezifikationen wurden weitere Standards entwickelt, die auf der Sicherungsschicht (ISO-OSI Layer 2) basieren. Hier sind die Standards 802.11c bis 802.11f sowie 802.11h bis 802.11j zu nennen.

3.2 Überblick

Die Tabelle 3.1 gibt einen Überblick über die momentan spezifizierten bzw. geplanten Standards der IEEE 802.11 Familie.

802.11	Ursprünglicher Standard, der 1997 verabschiedet wurde und dessen Datenrate bei 1 oder 2 MBit/s (brutto) liegt und auf dem 2,4 GHz Frequenzband operiert.
802.11a	Erweiterung der physikalischen Schicht von 1999, die die Nutzung des 5 GHz Frequenzbandes und eine Datenrate von bis zu 54 MBit/s (brutto) spezifiziert.
802.11b	Erweiterung der physikalischen Schicht von 1999, die auf dem 2,4 GHz Frequenzband basiert, aber eine Datenrate von 5,5 - 11 MBit/s (brutto) spezifiziert.
802.11c	MAC-Layer-Bridging gemäß 802.1d
802.11d	Anpassung an die regulatorischen Bestimmungen verschiedener Länder
802.11e	Unterstützung von QoS (Quality of Service)
802.11F	Interoperabilität zwischen Basisstationen
802.11g	Erweiterung der physikalischen Schicht von 2003, die auf dem 2,4 GHz Frequenzband basiert, aber eine Datenrate von bis zu 54 MBit/s (brutto) vorsieht.
802.11h	Reichweitenanpassung, Indoor- und Outdoor-Kanäle im 5 GHz Frequenzband
802.11i	Erweiterungen bezüglich Sicherheit und Authentifizierung

Fortsetzung siehe nächste Seite

802.11j	4.9 GHz - 5 GHz Operation in Japan
802.11k	Spezifikation, wie Access Points Informationen an Managementapplikationen auf den höheren ISO-OSI-Schichten weiterleiten.
802.11m	Eine Zusammenfassung früherer Ergänzungen, Bereinigungen von Fehlern aus vorausgegangenen Spezifikationen
802.11n	Neuer Standard, der noch in der Entwicklungsphase ist und Ende 2005 erscheinen soll. Hierbei ist eine Datenrate bis 540 MBit/s auf dem 5 GHz Frequenzband geplant.
802.11p	Erweiterung zum 802.11a Standard, der sich mit den Fahrzeug-zu-Fahrzeug-Netzen beschäftigt. Auch dieser ist noch in Entwicklung, geplant sind 27 MBit/s Datentransferrate bei 5,8 GHz. Geplant bis Ende 2008.
802.11r	Weitere Erweiterung die noch in der Entwicklung ist und sich mit Handovermechanismen beim Wechsel zwischen den Access Points beschäftigt.
802.11s	WMN (Wireless Mesh Network)
802.11t	WPP (Wireless Performance Prediction) legt u. a. ein Testverfahren fest
802.11u	Regelt das Zusammenspiel von 802.11-Netzen mit nicht-802-konformen Netzen
802.11v	Wireless Network Management

Tabelle 3.1: Auflistung momentan gültiger und geplanter IEEE 802.11 Standards

3.3 Physischer Layer

Aufgrund der Übertragungseigenschaften von elektromagnetischen Wellen in der Luft ist eine Übertragung der digitalen Signale im Basisband, wie sie bei drahtgebundenen Netzwerken in vielen auch sehr leistungsfähigen Standards Einsatz findet, nicht möglich.

Bei der Versendung analoger Signale mit Hilfe elektromagnetischer Wellen, wie z. B. bei herkömmlichen Radio- oder Fernsehsendern, werden die Audio- und Videosignale meist im Rahmen einer Amplituden- oder Frequenzmodulation übertragen. Bei der Übertragung digitaler Signale kommen die im Folgenden vorgestellten Technologien zum Einsatz.

3.3.1 FHSS - Frequency Hopping Spread Spectrum

Beim FHSS (dt. Frequenzspringen) handelt es sich um ein Frequenzspreizverfahren. Hierbei wird die verfügbare Bandbreite auf viele Kanäle mit kleinerer Bandbreite aufgeteilt. FHSS wird unterteilt in *Fast* und *Slow Hopping*. Generell wechselt hier die Trägerfrequenz frequentiv und diskret. Die Sequenz der Frequenzwechsel wird durch Pseudozufallszahlen bestimmt.

Für dieses Verfahren müssen zuerst die Nutzdaten schmalbandig moduliert werden, bevor sie dann durch einen zweiten Modulator durch den Frequenz-Synthesizer gespreizt werden. Auf der Empfängerseite ist wieder ein Frequenz-Synthesizer angeschlossen, der diese Spreizung rückgängig macht und danach die Daten konventionell demoduliert. Sender und Empfänger, die die gleiche Hopping-Sequenz benutzen, können sich untereinander verstehen.

Slow Hopping: Hierbei wird minimal 1 Bit pro Frequenzsprung übertragen, also z. B. 3 Bits bevor die Frequenz wechselt.

Fast Hopping: Hier wird maximal 1 Bit pro Frequenzsprung übertragen, es können aber durchaus auch 3 Frequenzsprünge innerhalb eines Bits stattfinden.

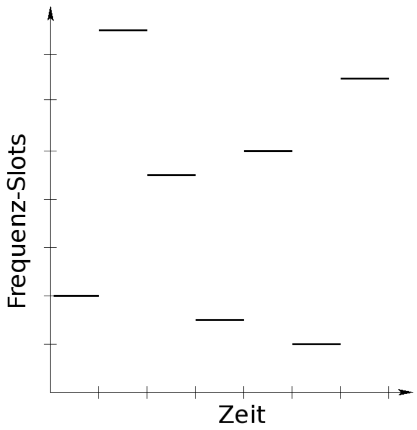

Abbildung 3.1: FHSS - Wechsel der Trägerfrequenz

Zwar ist FHSS leicht implementierbar, aber es nutzt nur einen schmalen Bereich des Spektrums zu einem Zeitpunkt und ist nicht so abhörsicher wie andere Verfahren (z. B. DSSS).

3.3.2 DSSS - Direct Sequence Spread Spectrum

Bei DSSS handelt es sich ebenfalls um ein Frequenzspreizverfahren. Die Idee hinter diesem Verfahren ist, eine vorgegebene Sequenz zu verwenden um ein Ausgangssignal zu spreizen. Diese Sequenz wird auch Chipping-Sequenz genannt und stellt eine Bitfolge dar.

Das Ausgangssignal wird nun mittels XOR mit der Chipping-Sequenz verknüpft. Übertragen wird das Ergebnis der XOR-Verknüpfung. Anders wie beim FHSS-Verfahren wird das Signal nun nicht nur auf verschiedenen Kanälen versendet, sondern durch den PN (Pseudo-Noise)-Code direkt von einem schmalbandigen in ein breitbandiges Signal umgewandelt und kontinuierlich versendet.

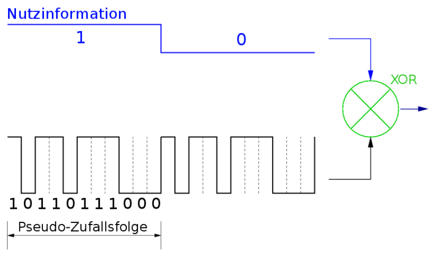

Abbildung 3.2: DSSS - Spreizung der Nutzdaten mittels XOR

DSSS ist je nach Chipping-Sequenz-Länge sehr gut gegen Abhören gesichert, das Signal geht für den Angreifer in ein Hintergrundrauschen über. Jedoch ist zu beachten, dass mit längerer Chipping-Sequenz mehr Bandbreite benötigt wird.

Das DSSS-Verfahren mit bis zu 11 MBit/s hat sich aufgrund der technischen Notwendigkeit nach Bandbreitenmaximierung durchgesetzt.

3.3.3 OFDM - Orthogonality Frequency Division Multiplexing

Im Gegensatz zu FHSS und DSSS ist OFDM ein Mehrträgerverfahren. Anstatt eines einzelnen Signalträgers werden mehrere Träger gleichzeitig verwendet. Jeder einzelne Träger ist phasen- und amplitudenmoduliert (ab 4 Bit pro Symbol) und trägt von daher die Information von mehreren Bits (typischerweise 2 bis 6 Bit) pro Symbol.

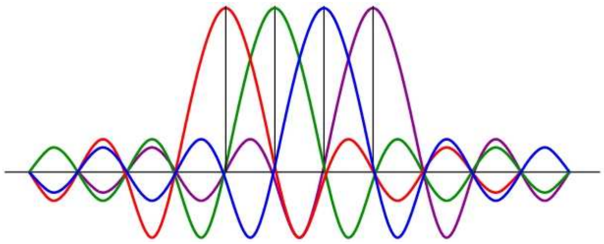

Abbildung 3.3: OFDM - Verteilung der Nutzdaten auf verschiedene Träger

Um sicherzustellen, dass sich die Trägerwellen nicht gleichzeitig beeinflussen, sind die Subträger zueinander orthogonal.

Anwendung findet dieses Verfahren im WLAN-Bereich bei den Standards 802.11g und 802.11a.

Um eine Datenrate von 54 MBit/s zu erreichen, wird für OFDM je Subcarrier als Modulationsverfahren 64-QAM (Quadrature Amplitude Modulation) angewendet. Für niedrigere Raten auch 16-QAM, QPSK (Quadrature Phase Shift Keying), BPSK (Binary Phase Shift Keying).

3.3.4 IR - Infrarot

Da dieses Verfahren nicht für den WLAN-Bereich relevant ist, aber trotzdem zur physischen Schicht gehört, soll es hier kurz Erwähnung finden.

Diese Technologie nutzt den IR-Frequenzbereich nahe des sichtbaren Lichtes. Es ist nur für Kurzstrecken geeignet (etwa 20 m). Das zu sendende Signal wird durch die Veränderung der Intensität des optischen Signals codiert, das durch die Variation der Stromstärke der LED des Senders realisiert wird. Auf der Empfängerseite arbeitet eine Photodiode, die in der Lage ist, anhand der Intensität und des dazu proportionalen Stroms, das Signal wahrzunehmen und zu verarbeiten. Für Infrarot-Systeme wird die Pulse Position Modulation verwendet.

3.3.5 Frequenzbänder (2,4 und 5 GHz)

Für die Datenübertragung über die Luft existieren in den IEEE 802.11 Standards momentan 2 Frequenzbänder. Zum einen das 2,4 GHz ISM (Industrial, Scientific, Medical)-Band und zum anderen das 5 GHz UNII (Unlicensed National Information Infrastructure)-Band. Beide Frequenzbänder sind für die lizenzfreie Nutzung durch die Bundesnetzagentur (BNA [5]) zugelassen. Die Standards 802.11, 802.11b und 802.11g setzen auf das 2,4 GHz ISM-Band, der 802.11a dagegen auf das 5 GHz UNII-Band. Die Vor- und Nachteile werden im folgenden Abschnitt genauer erläutert.

3.3.5.1 2,4 GHz Frequenzband

- Vorteile:
  - lizenzfreies Frequenzband
  - kein aufwändiges TPC (Transmitter Power Control)/DFS (Dynamic Frequency Selection) nötig, um die volle Sendeleistung auszuschöpfen
  - daher geringe Gerätekosten
  - folglich hohe Verbreitung
- Nachteile:
  - Frequenzband wird ebenfalls von anderen Geräten/Funktechniken (Bluetooth, Mikrowellenherde, Babyphones usw.) genutzt, daher mögliche Störungen
  - effektiv nur 3 parallel nutzbare Kanäle (Überlappung der Kanäle)

3.3.5.2 5 GHz Frequenzband

- Vorteile:
  - weniger genutztes Frequenzband, dadurch häufig störungsfreierer Betrieb möglich
  - in Deutschland 19 nicht überlappende Kanäle
  - höhere Reichweite, da mit 802.11h bis zu 1000 mW Sendeleistung möglich
- Nachteile:
  - geringe Verbreitung, dadurch höhere Anschaffungskosten
  - Störungen durch andere Geräte/Funktechniken (Radar)
  - nur Infrastruktur-Modus (siehe Abschnitt 3.7.2.2), aber kein Ad-Hoc-Modus (siehe Abschnitt 3.7.2.1) möglich

3.3.6 Sendeleistung

3.3.6.1 EIRP - Effective Isotropic Radiated Power

Das Produkt aus abgestrahlter Leistung der Sendeantenne und Gewinn der Sendeantenne bezogen auf einen Isotropstrahler wird als Effective Isotropic Radiated Power (EIRP) bezeichnet und hat heute besondere Bedeutung bei Betrachtungen zur Elektromagnetischen Verträglichkeit (EMV).

Als Einheit hat EIRP mW.

Rechnung in mW:

$$EIRP = \frac{PS \cdot GS}{L}$$

Rechnung in dB:

$$EIRP = PS + GS - L$$

mit:

PS: abgestrahlte Leistung der Sendeantenne
GS: Antennengewinn
L: Systemverlust (normalerweise gilt hier $L \geq 1$)

3.3.6.2 Maximale Sendeleistung in Deutschland

2,4 GHz Frequenzband In Deutschland ist es möglich 13 Kanäle des 2,4 GHz Frequenzbandes zu nutzen. Dabei ist aber zu beachten, dass es sich hierbei um überlappende Kanäle handelt.

Im Indoor- (In-Haus-) und Outdoor- (Freiland-)Bereich ist es erlaubt mit maximal 100 mW zu senden. Der Antennengewinn ist dabei zu berücksichtigen.

5 GHz Frequenzband WLANs nach dem Standard IEEE 802.11a dürfen mit der Erweiterung 802.11h in Deutschland sowohl innerhalb, als auch außerhalb geschlossener Räume aufgebaut werden. Die Regulierungsbehörde ließ eine entsprechende Verordnung Ende 2002 in Kraft treten. Innerhalb geschlossener Räume ist demnach das Band von 5,15 - 5,35 GHz für bis zu 200 mW sowie von 5,47 - 5,725 GHz für bis zu 1000 mW EIRP freigegeben. Für den Betrieb von WLANs außerhalb geschlossener Räume ist ausschließlich das Frequenzband von 5,47 - 5,725 GHz freigegeben, mit einer maximalen Sendeleistung von 1000 mW EIRP.

	Frequenzband in GHz	max. EIRP-Sendeleistung
Indoor	5,15 - 5,35	200 mW (23dBm)
	5,47 - 5,725	1000 mW (30 dBm)
Outdoor	5,47 - 5,725	1000 mW (30 dBm)

Tabelle 3.2: WLAN nach 802.11a/h: Übersicht Frequenzen und Sendeleistungen

Der Betrieb von WLANs nach 802.11a ohne 802.11h (also ohne TPC/DFS) ist in Deutschland nur im Frequenzband von 5,150 - 5,250 GHz mit stark herabgesetzten Sendeleistungen zugelassen.

	Frequenzband in GHz	max. EIRP-Sendeleistung
Indoor	5,150 - 5,250	60 mW
Outdoor	5,150 - 5,250	30 mW

Tabelle 3.3: WLAN nur nach 802.11a: Übersicht Frequenzen und Sendeleistungen

3.4 WLAN-Standards nach IEEE 802.11

Die Familie der IEEE 802 Standards bildet eine allgegenwärtige Plattform im Bereich lokaler Netze (engl. Local Area Networks kurz LAN). Dazu gehören unter anderem der Ethernet-Standard nach IEEE 802.3, Tokenring nach IEEE 802.5, Wireless nach IEEE 802.11 und weitere. Abbildung 3.4 zeigt die wichtigsten Standards und ihre wechselseitigen Beziehungen.

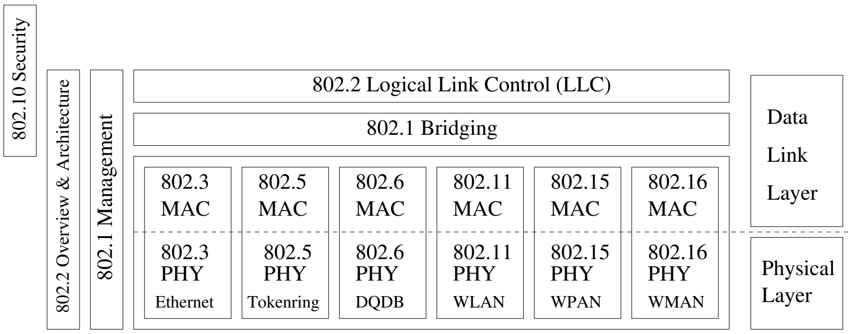

Abbildung 3.4: Die wichtigsten Standards der IEEE 802-Familie

Die IEEE 802 Standards beschreiben mit der Bitübertragungsschicht und der Sicherungsschicht die unteren zwei Schichten des ISO-OSI-Referenzmodells. Dabei wird die Sicherungsschicht noch einmal in *Media Access Control* und *Logical Link Control* unterteilt (siehe Abschnitt 4.2).

3.4.1 IEEE Standard 802.11

Im Jahr 1997 wurde der Urvater des WLAN standardisiert. Der IEEE 802.11 Standard spezifiziert die drahtlose Datenübertragung mit einer Datenrate von 1 und 2 MBit/s. Dabei unterteilt sich der Standard in die Beschreibung eines *Media Access Control*-Protokolls und drei *physical Layer*-Protokolle. Es werden dabei zwei Frequenzspreizverfahren (FHSS und DSSS) auf der Grundlage elektromagnetischer Wellen im 2,4 GHz ISM-Band beschrieben. Auch der Vorschlag der Datenübertragung über IR-Technologie ist enthalten, allerdings fand er nie praktische Anwendung.

3.4.2 IEEE Standard 802.11b

Bereits im September 1999 wurde ein weiterer Standard verabschiedet. Dieser basiert ebenfalls auf dem 2,4 GHz ISM-Band und ist zum ursprünglichen IEEE 802.11 Standard kompatibel. Hierbei werden Datenübertragungsraten von 5,5 und 11 MBit/s ermöglicht. Der 802.11b Standard setzt auf das weniger störanfällige Frequenzspreizverfahren DSSS. Die Erhöhung der Datenrate wird im Wesentlichen durch ein Modulationsverfahren mit verbesserter Nutzung des Frequenzspektrums erreicht. Durch den Einsatz von QPSK können mehr Bits pro Symbol übertragen werden als beim IEEE 802.11 genutzten BPSK.

3.4.3 IEEE Standard 802.11g

Die IEEE 802.11g-Arbeitsgruppe reichte ihren Entwurf im November 2001 ein, die Mitte 2003 verabschiedet wurde. Der g-Standard ist eine Erweiterung für den b-Standard und ist ebenfalls im 2,4 GHz ISM-Band angesiedelt. Statt der FHSS- und DSSS-Technologie wurde auf die OFDM-Technik gesetzt. Die Abwärtskompatibilität zu 802.11b wird durch die CCK (Complementary Code Keying)-Modulation sichergestellt. Durch die Kompatibilität mit dem 802.11b Standard ist es möglich, eine bestehende 802.11b Infrastruktur schrittweise auf den g-Standard umzustellen. Hiermit werden Datenraten zwischen 6 und 54 MBit/s bei etwa gleicher Reichweite erreicht.

3.4.4 IEEE Standard 802.11a

Im Unterschied zu den 802.11b/g-Standards arbeitet 802.11a im 5 GHz UNII-Band, wobei die Entwicklung hauptsächlich die Regeln für das US-amerikanische UNII-Band beachtet. Wie bereits im Abschnitt 3.3.5.2 beschrieben, wird dieses Frequenzband weniger von anderen Geräten genutzt. Der Standard 802.11a des IEEE stellt eine Erweiterung des grundlegenden Standards 802.11 dar. Dabei werden die Mechanismen des Kanalzugriffs auf der *Media Access Control*-Schicht unverändert übernommen. Der IEEE 802.11a Standard nutzt das OFDM-Verfahren. Um das Frequenzspektrum effektiv ausnutzen zu können, wird die Modulation (64-QAM, 16-QAM, QPSK, BPSK) an die aktuelle Signalqualität angepasst. Es sind Datenraten von 6 bis 54 MBit/s möglich.

Der Standard IEEE 802.11a wurde 1999 spezifiziert. Erste Produkte kamen im Jahr 2001 auf den Markt. Weltweit gibt es keine einheitliche Regelung für das 5 GHz Frequenzband sowie für die maximale Sendeleistung. Deshalb ist dieser Standard in Europa noch weniger verbreitet wie die Standards 802.11b/g.

Die Geräte des a-Standards sind wegen des unterschiedlichen Frequenzbandes nicht zu denen des b/g-Standards kompatibel. Es gibt aber mittlerweile Geräte, die sowohl b/g- als auch den a-Standard parallel anbieten.

3.5 Weitere WLAN-Standards

Es gibt neben den IEEE 802.11 Standards noch weitere Standards, die hier auch kurz Erwähnung finden sollen.

3.5.1 HIPERLAN - High Performance Radio Local Area Network

Die HIPERLAN Standards beschreiben eine alternative Technologie zum IEEE 802.11 Standard, um Rechner über ein Funknetz miteinander zu verbinden. Ab 1996 definierte eine Arbeitsgruppe namens BRAN (Broadband Radio Access Network) des ETSI (European Telecommunication Standards Institute) bis heute 4 HIPERLAN Standards:

- HIPERLAN/1 (1996) - 5 GHz, bis zu 23,5 MBit/s
- HIPERLAN/2 (2000) - 5 GHZ, bis zu 54 MBit/s
- HIPERACCESS (HIPERLAN/3) - 23 MBit/s und bis zu 5 km Reichweite (Letzte Meile)
- HIPERLINK (HIPERLAN/4) - feste Punkt-zu-Punkt-Verbindungen, 155 MBit/s und 150 m Reichweite

Ausführliche Informationen sind im Buch „*Mobile Computing*” [6] von Jörg Roth zu finden.

3.5.2 HomeRF - Home Radio Frequency

HomeRF wurde von der HomeRF Working Group entwickelt. Diese war ein Zusammenschluss von Mobilfunkunternehmen, in der u. a. Siemens und Motorola Mitglieder waren. Anfang 2003 wurde diese Arbeitsgruppe aufgelöst, als der IEEE 802.11b Standard benutzbar wurde.

HomeRF ist ein Funknetzstandard, der für Privathaushalte und kleinere Büros entwickelt wurde. Es nutzt das lizenzfreie 2,4 GHz ISM-Band und liefert maximal 10 MBit/s (in einer späteren Version 20 MBit/s).

Als Anwendungsgebiet wurden Fernsehgeräte und HiFi-Geräte vorgesehen, die Audio und Videos über etwa 50 m übertragen sollten. Hauptsächlich sollte es einen gemeinsamen Internet-Zugang für mehrere PCs bieten.

Durch die weite Verbreitung der IEEE 802.11 Standards und deren breite Unterstützung ist der HomeRF Standard allerdings nicht sehr stark verbreitet und genutzt worden.

Eine ausführlichere Dokumentation dieses Standards ist im Kapitel 7 im Buch „*Wireless LAN*" [7] von Axel Sikora beschrieben.

3.5.3 Bluetooth

Die Entwicklung von Bluetooth wurde 1994 durch die Firma Ericsson gestartet. Zusammen mit anderen Industriepartnern wurde 1998 die Bluetooth SIG (Bluetooth Special Interest Group) gegründet, um Bluetooth als de facto-Standard zu etablieren. Inzwischen ist Bluetooth in der Arbeitsgruppe der IEEE für WPANs (Wireless Personal Area Network) als IEEE Standard 802.15.1 adaptiert worden.

Ein Ziel der Bluetooth-Entwicklung ist es, die verschiedensten Geräte ohne Konfiguration durch den Benutzer miteinander kommunizieren zu lassen.

Bluetooth nutzt ebenfalls das lizenzfreie 2,4 GHz ISM-Band und FHSS, um eine Robustheit gegenüber Störungen zu erreichen.

Theoretisch kann eine Datenübertragungsrate von 723,2 kBit/s Downstream bei gleichzeitigen 57,6 kBit/s Upstream erreicht werden. Mit der Version 2.0 können Daten mit rund 2,2 MBit/s übertragen werden. Seit Version 1.1 kann ein Bluetooth-Gerät gleichzeitig bis zu sieben Verbindungen aufrechterhalten. Da es sich, wie bei allen anderen WLANs, um ein „Shared Medium" handelt, müssen sie sich die Bandbreite teilen.

Die Reichweite von Bluetooth beträgt nur wenige Meter (etwa 10 bis 100 m).

Bluetooth ist mittlerweile ein weit verbreiteter Standard und kommt u. a. in Handys, schnurlosen Tastaturen/Mäusen und Headsets vor.

Bluetooth wird ausführlicher in „*Grundkurs Mobile Kommunikationssysteme*" [8] von Martin Sauter und in „*Digitale Funkkommunikation mit Bluetooth*" [9] von Merkle und Terzis behandelt.

3.6 Antennen

Für den Einsatz von WLAN sind die Antennen eines der maßgeblichen Kriterien. Entscheidend sind dabei die Form der Wellenausbreitung und das Maß der Verstärkung.

Man unterscheidet grundsätzlich zwei Ausbreitungsmuster: omnidirektionale und direktionale. Omnidirektionale Antennen senden kreisförmig in horizontaler Richtung, d. h. der Empfang ist dann am besten, wenn man sich auf derselben Höhe befindet und am schlechtesten direkt unter der Antenne. Direktionale Antennen ermöglichen, wie der Name schon sagt, die Ausrichtung der Wellen in einer bestimmten Richtung.

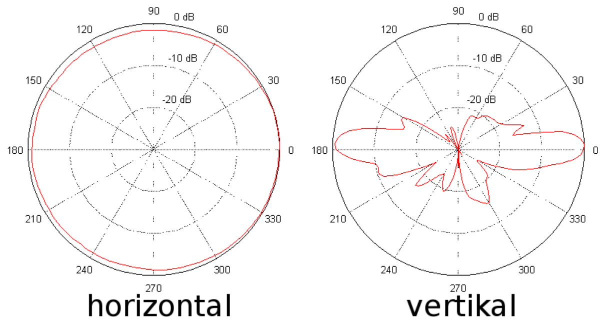

Abbildung 3.5: Elektromagnetisches Ausbreitungsmuster einer omnidirektionalen Antenne

Ein wesentlicher Parameter, der die Reichweite elektromagnetischer Wellen bestimmt, ist deren Leistung. Die Feldstärke einer elektromagnetischen Welle nimmt im freien Raum umgekehrt proportional mit der Entfernung zum Sender ab. Die Eingangsleistung am Empfänger nimmt entsprechend mit dem Quadrat der Entfernung ab.

In der Praxis werden Antennen eingesetzt, die ihre Leistung in bestimmte Vorzugsrichtungen bündeln. Antennen werden in zahllosen Varianten angeboten. Da diese unabhängig vom verwendeten Protokoll und vom Hersteller der aktiven Geräte sind, sind bei der Auswahl der Antenne einige wesentliche Parameter zu beachten. Diese sind zum einen die Montage (Wand, Mast, PC usw.), die Richtungscharakteristik (z. B. omnidirektional, vertikal, horizontal), die Bauform (Omni, Sektor, Yagi, Parabol usw.) sowie der Gewinn der Antenne. Zu beachten ist die Dämpfung des Antennenkabels, da diese mindestens 0,2 dB/m beträgt.

3.7 WLAN-Architektur

3.7.1 Grundbegriffe

Um ein WLAN näher zu beschreiben, werden verschiedene Begriffe verwendet. Dieses Kapitel soll die Bedeutung der einzelnen Begriffe erläutern.

- **Wireless Medium** Ein drahtloses Medium (Luft) dient zur Übertragung von Daten. Für die Übertragung in diesem Medium sind im IEEE 802.11 Standard verschiedene Technologien und Standardisierungen definiert.
- **Distribution System** Man spricht hier vom „Verteilungs-Netzwerk” (auch „Backbone Netzwerk” genannt) und meint damit ein Netzwerk, das die Access Points miteinander oder mit anderen Netzwerken verbindet. Der IEEE 802.11 Standard spezifiziert dafür keine spezielle Technologie. Eine einfache ISO-OSI Layer-2-Verbindung (z. B. Ethernet oder WLAN selbst (sogenannter „Bridge Mode”)) reicht dafür bereits aus.

- **Access Point** Hierbei handelt es sich um Netzwerkkomponenten, deren Hauptaufgabe ist es, die „Funkwelt” mit der „Kabelwelt” zu verbinden. Sie verbinden zwei verschiedene Netzwerktechnologien, man bezeichnet dies auch als „Bridge”.
- **Station** Wird auch oft als WLAN-Client bezeichnet und meint ein Computer-System mit einer WLAN-Netzwerkkarte (engl. Network Interface Card kurz NIC). Typische Geräte sind Notebooks/Laptops, Handhelds usw.
- **Association** Unter Association versteht man das Bekanntmachen einer Station beim Access Point. Zur Station werden im Access Point entsprechende Informationen gespeichert.
- **Disassociation** Die Disassociation dient zum Trennen der WLAN-Verbindung. Dazu wird eine Nachricht („lösche alle Informationen zu dieser Station”) gesendet, worauf allerdings keine Antwort erwartet wird. Sie kann weder vom Access Point noch von einer mobilen Station abgelehnt werden.
- **Reassociation** Mechanismus zum erneuten Beitritt nach einer erfolgten Abmeldung an einem Access Point.
- **Authentication** Association nach Identitätsnachweis.
- **Deauthentication** Beendung der Authentication.
- **SSID** Service Set Identity wird auch oft als Netzwerkname bezeichnet. Es ist ein Identifikator für das WLAN-Netzwerk. Im BSS (siehe Abschnitt 3.7.2.2) und ESS (siehe Abschnitt 3.7.2.3) ist die SSID immer gleich. Sie darf maximal 32 Bytes lang sein und Zahlen und Buchstaben enthalten.
- **Beacon** Wird durch den Access Point in regelmäßigen Intervallen gesendet. Es gibt die Existenz des Netzwerkes an alle „Zuhörer” im Empfangsbereich bekannt. Neben verschiedenen Parametern des WLANs wird auch die SSID bekannt gegeben.

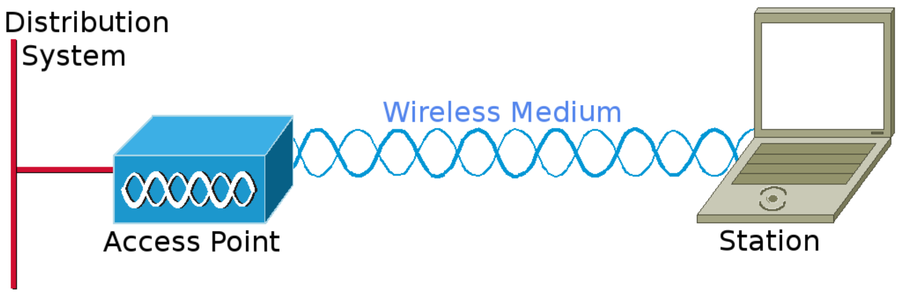

Abbildung 3.6: Komponenten eines 802.11 Netzwerkes

3.7.2 WLAN-Technologien

3.7.2.1 IBSS - Independent Basic Service Set

Ein IBSS (Independent Basic Service Set), besser bekannt als „**Ad-Hoc**”-Netzwerk, besteht aus zwei oder mehr drahtlosen Stationen, die direkt miteinander kommunizieren, ohne die Anwesenheit eines Access Points. Es wird auch oft als *Peer-to-Peer*-Netzwerk bezeichnet. Dies kann einfach und ohne großen Aufwand aufgebaut werden. Dabei wird eine Station ausgewählt, um als Steuerungs- und Basis-Station zu agieren und bietet so eine ähnliche Funktion wie ein Access Point, nur dass es hierbei keine Verbindung zum Kabelnetz gibt.

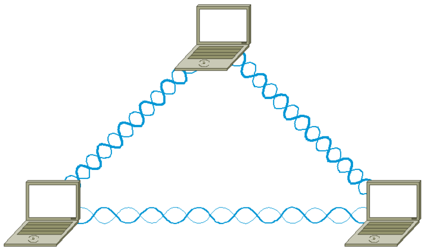

Abbildung 3.7: IBSS - Independent Basic Service Set

3.7.2.2 BSS - Basic Service Set

In einem BSS (Basic Service Set) muss zwingend ein Wireless Access Point vorhanden sein, man nennt dies auch „**Infrastruktur Modus**” („Netzwerk mit Zugriffspunkt”). Ein Access Point hat mehrere Aufgaben. Er ist eine Basis Station und eine Netzwerkzugangssteuerung (engl. „network access controller”) für die mobilen Stationen im BSS. Er stellt auch gegebenfalls für die mobilen Stationen im BSS die Verbindung ins Kabel-Netzwerk (LAN) her. Ein Access Point kann auch den Zugang zum Netzwerk steuern/blockieren, indem er z. B. anhand von Listen mit Hardware-Adressen den Zugang gewährt oder blockiert.

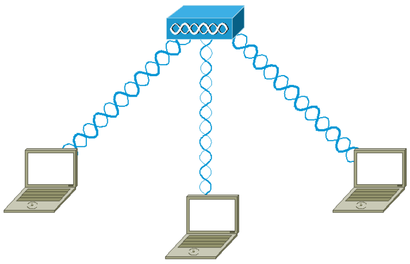

Abbildung 3.8: BSS - Basic Service Set

3.7.2.3 ESS - Extended Service Set

Ein ESS (Extended Service Set) ist eine Serie von zwei oder mehr Basic Service Sets, die mit einem LAN verbunden sind. Jeder Access Point stellt für seine BSS-Teilnehmer eine Verbindung zum LAN her. Jedes BSS identifiziert sich mit einer einmaligen Nummer, der BSS-ID (die MAC-Adresse des Access Points). Die einzelnen mobilen Stationen in einem ESS suchen sich automatisch den Access Point (BSS), zu dem sie die beste Verbindung haben. Wenn kein Access Point gefunden wird, scannt das Gerät solange bis es einen benutzbaren Access Point gefunden hat.

Ein ESS kann so konfiguriert werden, dass sich die einzelnen mobilen Stationen in der Reichweite von beliebigen Access Points bewegen können. In diesem Fall teilen sich die Stationen ein gemeinsames ESS, dass durch eine ESS-ID (ESSID) identifiziert wird.

Ein Access Point kann auch dazu benutzt werden, um ein WLAN zu teilen. Unter diesen Umständen muss mehr als ein ESS benutzt werden, zwei oder mehr einzelne Extended Service Sets können am gleichen Ort aufgebaut sein, wobei jede Station im WLAN immer nur ein ESS davon benutzen kann.

Auch beim ESS spricht man vom „**Infrastruktur Modus**".

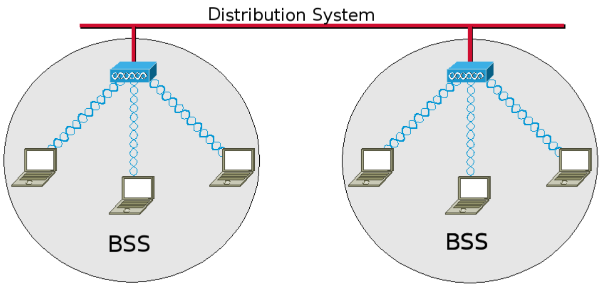

Abbildung 3.9: ESS - Extended Service Set

3.7.3 Datentransport mittels IEEE 802.11

Die Übertragung der Daten durch die Luft besitzt gegenüber drahtgebundener Datenübertragung einige Besonderheiten, die beachtet werden müssen. So kann es u. a. passieren, dass gleiche Datenpakete durch Reflektion zu verschiedenen Zeitpunkten beim Empfänger eintreffen. Außerdem handelt es sich um ein „Shared Medium" in dem immer nur ein Kommunikationspartner gleichzeitig senden kann.

Ein weiterer wichtiger Faktor den es zu beachten gilt, ist die sich ständig ändernde Qualität der Verbindung. Alle diese Faktoren sind im IEEE 802.11 Standard beachtet.

Die im IEEE 802.11 Standard angewandten Verfahren sind in den Kapiteln 3 und 4 im Buch „*802.11 Wireless Networks*" [10] von Matthew Gast beschrieben.

4 Verschlüsselung

4.1 Hintergrund

Beim drahtgebundenen (engl. wired) Netzwerk werden die Daten entweder über Coax-Kabel, TP oder LWL (Lichtwellenleiter, engl. Fiber) übertragen. Um Zugang zu den Daten zu bekommen, benötigt ein Angreifer physischen Zugang zum Netzequipment (Kabel, Switch, Router usw.). Auch wenn dies kein ausschlaggebender Punkt für Netzsicherheit ist, so bietet dies zumindest soweit Schutz, wie man diese physikalische Schicht (Bitübertragungsschicht) absichern kann.

Ganz anders sieht es bei Funknetzen aus: WLANs basieren auf der Übertragung von Funkwellen im freien Raum. Eine räumliche Eingrenzung der Funkwellenausbreitung ist kaum möglich. So durchdringen sie Wände, Türen, Fenster, überqueren Strassen, Wege und dringen so auch bis in andere Gebäude vor. Man kann hier von einer unkontrollierbaren Ausbreitung sprechen.

Durch das unsichere Medium Luft kann ein Angreifer selbst aus weiter Entfernung ohne Zugang zum eigentlichen Netzequipment Daten ausspionieren.

Im Kapitel 3 wurden bereits die speziellen Frequenzspreizverfahren bzw. -modulationsverfahren (FHSS, DSSS und OFDM) für die Datenübertragung über das WLAN beschrieben. Diese Verfahren bieten aber keinen wirksamen Schutz gegen das Abhören und Manipulieren der übertragenen Daten.

Auch die Wahl einer geheimen Netzwerk-ID oder der Einsatz von speziellen ACLs (Access Control Lists) bieten nicht ausreichend Schutz.

4.2 Allgemein

Zweck einer Verschlüsselung ist das Verhindern von möglichem Abhören oder Manipulieren der übertragenen Daten durch Dritte. Hierzu zählen hauptsächlich:

- Unterbrechung/Behinderung der Verbindung
- Abhören der Verbindung
- Datenmodifikation
- Identitätsfälschung

- Behinderung des Empfängers/Senders

Aspekte dieser Betrachtung sind also:

- Vertraulichkeit
- Datenintegrität
- Korrektheit der Authentifizierung und Autorisation

In diesem Kapitel soll es um die Verschlüsselung der Daten (Vertraulichkeit) und die Sicherung der Datenintegrität gehen. Authentifizierung und Autorisation werden im nächsten Kapitel genauer untersucht.

Verschlüsselung und Integritätssicherung können an den verschiedene Schichten des ISO-OSI-Modells (Abbildung 4.1) realisiert werden.

7	Anwendung
6	Darstellung
5	Kommunikation
4	Transport
3	Vermittlung
2	Sicherung
1	Bitübertragung

Abbildung 4.1: ISO-OSI-Modell

Die IEEE 802 Standards beschreiben die untersten Schichten des ISO-OSI-Modells. Die Tabelle 4.1 gibt einen Überblick über die wichtigsten dieser Standards.

802.1	Bridging and Management
802.2	Logical Link Control
802.3	CSMA/CD (Carrier Sense Multiple Access With Collision Detection) Access Method
802.4	Token-Passing Bus Access Method
802.5	Token-Ring
802.7	Broadband LAN
802.11	Wireless
802.14	Cable Modem

Tabelle 4.1: IEEE 802 Standards

Die IEEE 802 Standards spalten die Sicherungsschicht in zwei Teile: LLC (Logical Link Control) und MAC (Media Access Control).

Der IEEE 802.2 Standard definiert eine gemeinsame Teilschicht (LLC), die durch die MAC-Teilschicht und die physische Schicht (Bitübertragungsschicht) benutzt wird.

Der am häufigsten benutzte IEEE 802 Standard ist 802.3 (CSMA/CD Access Method), auch als Ethernet bekannt.

Der IEEE 802.11 Standard definiert die Sicherungs- und die physische Schicht für Funknetze. Dabei ist der *Media Access Control*-Teil für den Transport zwischen LLC und physischer Schicht zuständig und bildet eine einheitliche Schnittstelle für die verschiedenen Möglichkeiten der Bitübertragung im Funknetz (IEEE 802.11a/b/g).

Die Abbildung 4.2 zeigt die Untergliederung der Schichten 1 und 2 des ISO-OSI-Modells für IEEE 802.11:

2	Sicherung	802 LCC			
		802.11 MAC			
1	Bitübertragung	802.11	802.11b	802.11a	802.11g

Abbildung 4.2: ISO-OSI Layer 1 und 2

4.3 Die Sicherungsschicht - IOS-OSI Layer 2

Für die Verschlüsselung der Verbindung auf der Sicherungsschicht sieht der IEEE 802.11 Standard den *Media Access Control*-Teil vor. Im Zuge der Weiterentwicklung wurden hierfür verschiedene Verfahren entwickelt und standardisiert. Abbildung 4.3 zeigt die Einordnung dieser Verfahren, die nachfolgend näher beschrieben werden.

2	802 LCC		
	WEP	TKIP	CCMP
1	Bitübertragung		

Abbildung 4.3: *Media Access Control*-Verschlüsselungsverfahren

4.3.1 WEP - Wired Equivalent Privacy

WEP ist ein Verschlüsselungsverfahren für WLAN, das bereits im IEEE 802.11 Standard integriert wurde. Aus dem Namen ist abzulesen, dass man hiermit versucht, die Sicherheit eines drahtgebundenen Netzes zu erreichen.

Es wurde entwickelt, um das Abhören der Funkverbindung zu verhindern, die Integrität der Daten zu sichern und als Zugangskontrolle zu fungieren, was bei der Entwicklung aber eher sekundär war.

Generell handelt es sich bei der Verschlüsselung um eine einfache XOR-Verknüpfung der Nutzdaten (Bitstrom) mit einem, aus dem RC4-Algorithmus generierten, pseudozufälligen Schlüsselstrom (engl. keystream).

4.3.1.1 RC4

RC4 ist ein sogenannter Strom-Chiffrierer, der 1987 von Ronald L. Rivest entwickelt wurde und bis 1994 geheim war. RC4 kann mit einer variablen Schlüssellänge bis 2048 Bits arbeiten und verschlüsselt immer ein Byte auf einmal. Die Codierung mit geringer Schlüssellänge gilt als unsicher und leicht entschlüsselbar.

Eine ausführliche Abhandlung der Funktionsweise und des zugrunde liegenden Algorithmus ist im Buch „*Angewandte Kryptographie*" [11] von Bruce Schneier zu finden.

4.3.1.2 Funktionsweise von WEP

Das Verfahren beruht auf einem gemeinsamen geheimen Schlüssel (engl. shared secret key), der den Kommunikationspartnern bekannt ist. Neben dem Schlüssel gibt es noch einen 24 Bit großen IV (Initialisierungsvektor), der für jede zu schützende Nachricht gebildet wird. Die Verknüpfung aus IV und dem geheimen Schlüssel dient als Eingabe für den Pseudozufallsgenerator, der daraus einen *Keystream* erzeugt.

Zu jeder Nachricht wird mittels Integritätsprüfverfahren (CRC-32) ein sogenannter ICV (Integrity Check Value) berechnet und an die eigentliche Nachricht angehängt. Die daraus resultierende Nachricht wird durch eine XOR-Verknüpfung mit dem *Keystream* aus dem RC4 verbunden und damit zum Ciphertext (dt. verschlüsselte Nachricht).

Gesendet werden nur der unverschlüsselte Initialisierungsvektor, eine Key-ID und der Ciphertext. Abbildung 4.4 zeigt den Codierungsvorgang beim Sender und Abbildung 4.5 die Decodierung beim Empfänger.

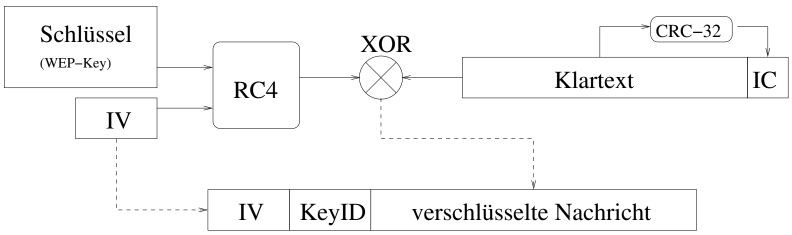

Abbildung 4.4: Datenverschlüsselung mittels WEP

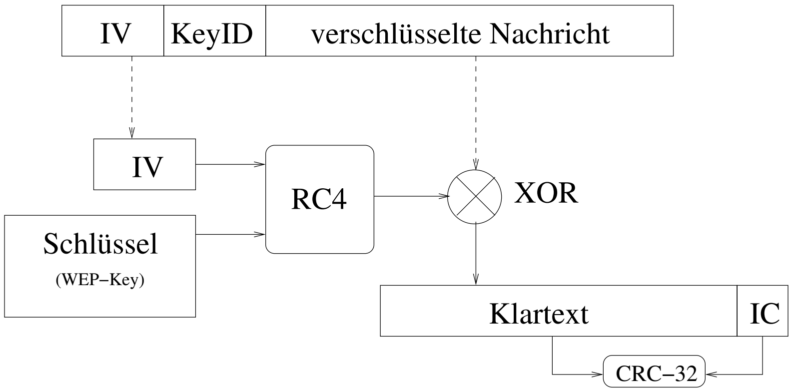

Abbildung 4.5: Datenentschlüsselung mittels WEP

4.3.1.3 Authentifizierung

Man unterscheidet beim WEP zwei Authentifizierungsverfahren:

Open System Authentication - Dieses System ist die Standard-Authentifizierung. Dabei werden alle Stationen für das WLAN freigeschaltet und es findet praktisch keine Authentifizierung statt.

Shared Key Authentication - Die zweite Variante gilt als die „sicherere”. Die Authentifizierung erfolgt dabei über ein Challenge-Response-Verfahren mit einem geheimen Schlüssel (WEP-Key).

4.3.1.4 Schwachstellen

Bei WEP gibt es verschiedene Schwachstellen. Die erste Schwachstelle ist der, mit 24 Bit viel zu kleine, IV. Ein IV wiederholt sich bereits nach einigen Stunden. Durch Übertragung bekannter Nachrichten, wie z. B. beim SMB (Server Message Block)-Dienst von Microsoft, kann eine Entschlüsselung vereinfacht werden.

Die Verschlüsselung durch RC4 mit einem 64 bzw. 128 Bit Schlüssel gilt mathematisch als unsicher. Da der IV 24 Bit lang ist, ist der geheime gemeinsame Schlüssel von WEP 40 bzw. 104 Bit lang.

Weiterhin sind durch die CRC-32-Eigenschaften Änderungen an den Daten ohne Inhaltskenntnis möglich. So kann ein Angreifer ein WEP-Paket manipulieren und dabei relativ einfach sicherstellen, dass auch die Prüfsumme, also der ICV, stimmt.

Eine nicht unwesentliche Sicherheitslücke stellt hierbei die Organisation der Schlüsselverteilung dar. Diese wurde im Standard nicht spezifiziert. Allein durch geringe Unachtsamkeit, Diebstahl usw. kann der geheime Schlüssel entwendet werden und muss somit, um die Sicherheit der Datenübertragung zu gewährleisten, neu vergeben und verteilt werden. Durch eine große Anzahl von Nutzern verringert sich die Geheimhaltung dieses Schlüssels und erhöht den Aufwand bei der Verteilung eines neuen.

4.3.2 WPA - Wi-Fi Protected Access

WPA ist ebenfalls eine Verschlüsselungmethode für WLAN. Allerdings ist sie kein offizieller IEEE Standard, sondern stammt von Wi-Fi, einer Organisation von verschiedenen Firmen, die eine Teilmenge des IEEE 802.11i Standards vorwegnahm. Der Grund dafür lag in der Unzufriedenheit über die mangelnde Sicherheit des WEP und Verzögerungen bei der Verabschiedung des IEEE 802.11i Standards.

Wie bereits WEP basiert auch WPA auf dem RC4 Stromchiffre. Der Unterschied zu WEP besteht aber nicht nur darin, dass ein 48 Bit Initialisierungsvektor verwendet wird, sondern auch in der „per-packet key mixing"-Funktion. Für die Datenintegrität wird auf MIC (Message Integrity Check) gesetzt.

Durch ein spezielles Schlüsselmanagement werden dynamisch Schlüssel für die Verschlüsselung mittels TKIP (Temporal Key Integrity Protocol) und Datenintegritätssicherung mittels MIC generiert, die auch während der Sitzung durch einen sogenannten „*Re-Keying*"-Mechanismus erneuert werden können.

Für die Authentifizierung von Nutzern bietet WPA zwei Möglichkeiten, zum einen über PSK (pre-shared key; gemeinsamer geheimer Schlüssel) und zum anderen mittels EAP über 802.1x (Beschreibung im Kapitel 5).

4.3.2.1 MIC - Message Integrity Check

Mit dem MIC, auch *„Michael"* genannt, kann eine kryptographische Prüfsumme zur Sicherstellung der Integrität von Nachrichten gebildet werden. Dieses Verfahren verwendet zwei Mechanismen, die *Sequence Number* mit 48 Bit Länge und den *Frame Integrity Check*. Bei der *Sequence Number* wird ein Zähler zum Datenpaket hinzugefügt. Datenpakete außerhalb der Reihenfolge werden verworfen. Beim *Frame Integrity Check* handelt es sich um ein Prüfsummenverfahren. MIC hat insgesamt 64 Bit, einen geringen Rechenaufwand und wird zusammen mit dem Datenpaket verschlüsselt übertragen.

4.3.2.2 TKIP - Temporal Key Integrity Protocol

Wie der Name sagt, handelt es sich dabei um eine Zwischenlösung, die nur übergangsweise bis zur Einführung eines wirklich starken Verschlüsselungsverfahrens genutzt werden soll, aber trotzdem mit den Problemen von WEP aufräumt. Eine Anforderung an das Design des Verschlüsselungsverfahrens war demzufolge, dass dies auch auf bereits existierender WEP/RC4-Hardware mit vertretbarem Aufwand implementierbar sein sollte.

Da bereits bei der Entwicklung des TKIP eine Datenrate von bis zu 540 MBit/s (IEEE 802.11n) abzusehen war, finden sich im TKIP viele von WEP bekannte Komponenten wieder, die in WLAN-Karten üblicherweise in Hardware existieren und damit effektiv für TKIP genutzt werden können.

TKIP kümmert sich um die Berechnung der eigentlichen Schlüssel für die RC4-Engine. Im Gegensatz zu WEP wird der eigentliche Schlüssel und der im Paket enthaltene IV niemals direkt als RC4-Schlüssel genutzt. Er durchläuft zusammen mit dem IV zwei sogenannte *Key-Mixing*-Phasen. Ein Angreifer kann also vom IV keine direkten Rückschlüsse auf den RC4-Schlüssel ziehen, was das Problem der „schwachen" IVs von WEP löst. Das *Key-Mixing* selbst ist so ausgelegt, dass niemals schwache RC4-Schlüssel entstehen können.

Der IV wird kontinuierlich inkrementiert und im Klartext in das zu übertragene Paket integriert. Allerdings ist der IV hier statt 24 jetzt 48 Bit lang. Damit kann ein Sender jetzt etwa 280 Billionen Pakete übertragen, bevor der 128 Bit lange TKIP Schlüssel zwingend gewechselt werden müsste.

Der IV wird aus Optimierungsgründen zweigeteilt: in einen 16 Bit langen *Lo*-Teil und einen 32 Bit langen *Hi*-Teil. Der Hintergrund dafür ist, dass das *Key-Mixing* in zwei Phasen abläuft:

- Für die erste (rechenintensive) Phase wird nur der *Hi*-Teil benötigt. Sie muss nur alle 2^{16} Pakete durchlaufen werden.
- Die zweite, relativ einfache Phase des *Key-Mixing* benutzt die Ausgabe der ersten Phase sowie den *Lo*-Teil des IV, um den eigentlichen RC4-Schlüssel zu erstellen.

Da der Initialisierungsvektor von Paket zu Paket streng monoton ansteigt, muss der Empfänger die Phase 1 ebenfalls nur alle 2^{16} empfangenen Pakete durchlaufen. Der Entschlüsselungsteil von TKIP muss die monotone Reihenfolge des IVs überprüfen und Pakete verwerfen, die einen bereits benutzten IV enthalten. Somit ist es z. B. möglich Replay-Attacken zu verhindern.

Als weiteres Detail mischt TKIP in der ersten Phase auch noch die Hardware-Adresse des Senders ein. Auf diese Weise ist sichergestellt, dass eine Verwendung gleicher IVs von verschiedenen Sendern nicht zu identischen RC4-Schlüsseln und damit wieder zu Angriffsmöglichkeiten führt.

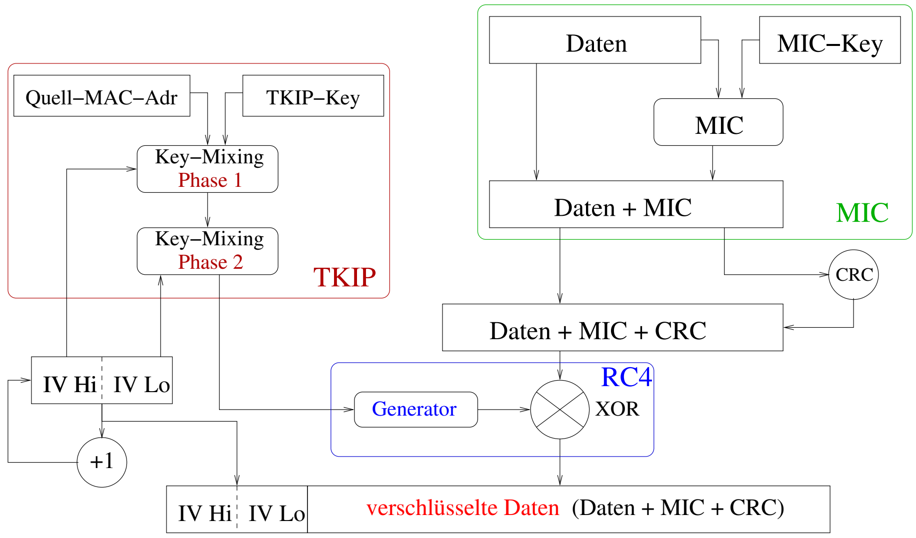

Abbildung 4.6: WPA - TKIP/MIC Funktionsweise

4.3.2.3 Schlüsselmanagement

Für die verschlüsselte Übertragung der Daten ist es wichtig, dass beide Seiten, also Station und Access Point, den gleichen Key benutzen. Anders als beim WEP werden aber keine statischen, sondern dynamisch erzeugte Schlüssel benutzt. Hierfür wurde der sogenannte *4-Way-Handshake* im 802.11i Standard entwickelt und bereits vor Veröffentlichung des Standards schon im WPA implementiert.

Dazu wird auf die Möglichkeit von 802.1x (siehe Kapitel 5) aufgesetzt. Dabei wird sowohl auf der Station als auch auf dem Access Point ein *Pairwise Master Key* installiert. Der Access Point bekommt diesen vom Authentifizierungsserver (siehe Kapitel 6) mitgeteilt, die Station und der Authentifizierungsserver generieren diesen Schlüssel während der Authentifizierung. Da diese aber nicht Teil des IEEE 802.11i-Standards ist, wird hier auf eine detaillierte Beschreibung des Ablaufs verzichtet.

Der zweite Schritt ist der *Key-Handshake*, der sich dabei in zwei Phasen gliedert, erst der *Pairwise Key Handshake* und dann der *Group Key Handshake*. Im ersten Schritt tauschen Access Point und Station ein Paar Zufallszahlen („Nonces”) aus. Nun wird mittels Hashverfahren aus dem *Pairwise Master Key* und den Zufallszahlen der temporäre *Pairwise Transient Key* erzeugt. Dieser Schlüssel ist für die Verwendung im TKIP 512 Bit lang. Dabei wird der Schlüssel in mehrere jeweils 128 Bit lange Teilschlüssel aufgespalten, die für die Verschlüsselung (z. B. *TKIP-Key*) und die Integritätssicherung (*z. B. MIC-Key*) genutzt werden.

Zuletzt wird noch der *Group-Key* ausgetauscht mit dem Multicast und Broadcast codiert werden. Der Schlüsselaustausch erfolgt mit dem eben erzeugten temporären Schlüssel. Der *Group-Key* wird vom Access Point festgelegt und an allen Stationen installiert. Nachdem die Installation der Schlüssel abgeschlossen ist, schaltet der Access Point die Station für den normalen Datenverkehr frei. Ein sogenanntes *Re-Keying* kann von beiden Seiten aus ausgelöst werden (z. B. bei zwei falschen MICs).

Da der Master-Key nie direkt für die Datenübertragung genutzt wird, kann er später weiter verwendet werden.

Falls statt dem 802.1x das PSK-Verfahren eingesetzt wird, wird der *Pairwise Master Key* anhand des gemeinsamen geheimen Schlüssels über ein Hashverfahren in Verbindung mit der SSID erzeugt. Danach erfolgt die Generierung des *Pairwise Transient Key* und der Austausch des *Group-Key* genau wie bereits oben beschrieben.

Für ältere Access Points gibt es noch verschiedenste Kompatibilitätsmodi für den Schlüsselaustausch.

4.3.3 IEEE 802.11i

Wie bereits im Abschnitt 4.3.2 beschrieben, ist WPA nur ein Vorgriff auf die erst später folgende Spezifizierung des IEEE 802.11i Standards (von der Wi-Fi Alliance WPA2 genannt). Der Standard wurde im Juni 2004 verabschiedet. Die Gründe für die Einführung dieses Standards sind die Schwachstellen im WEP, so wie sie bereits in den beiden vorangegangenen Abschnitten beschrieben wurden.

In großen Teilen nutzen WPA und WPA2 identische Protokolle und Verfahren. WPA ist eine Untermenge des gesamten 802.11i Standards. Folgende Funktionen wurden im IEEE 802.11i spezifiziert, die in WPA nicht beachtet wurden:

- sicheres IBSS (geschützter Ad-Hoc-Modus)
- Secure Fast Handoff (schnelles Roaming) und Pre-Authentication (frühzeitige Authentifizierung)
- sichere Abmeldung vom Netz („Secure De-Authentication and disassociation")
- erweiterte Verschlüsselungsprotokolle wie AES-CCMP (Advanced Encryption Standard - Counter Mode with Cipher Block Chaining Message Authentication Code Protocol)

Weil WPA eine Untermenge des 802.11i Standards ist, kann der 802.11i als abwärtskompatibel bezeichnet werden. Die meisten Funktionen lassen sich durch ein Firmware-Upgrade der WLAN-Netzwerkkarte nachrüsten, allerdings wird das integrierte Verschlüsselungsprotokoll AES-CCMP nicht mehr auf alter bzw. nicht ausreichend leistungsstarker WLAN-Hardware laufen.

802.11i enthält eine völlig neue Sicherheitsarchitektur, die maximale Sicherheit für WLANs gewährleisten soll. Von der TGi (Task Group i) der IEEE wurde diese neue Sicherheitsarchitektur RSN (Robust Security Network) getauft.

4.3.3.1 Kryptographische Funktionen

- AES-CCMP benutzt einen 48 Bit IV (Initalisierungsvektor)
- Prüfsumme mittels MIC (MIC wird AES verschlüsselt übertragen)
- temporäre, abgeleitete Schlüssel werden verwendet
- automatische Neuverschlüsselung (*Re-Keying*)
- verwendet das *Packet Number*-Feld (48 Bit) um vor Replay-Angriffen zu schützen

4.3.3.2 Schlüsselmanagement

Das Schlüsselmanagement von WPA2 wurde bereits bei WPA benutzt und im Abschnitt 4.3.2.3 beschrieben. Für CCMP wird allerdings nur ein 384 Bit langer *Pairwise Transient Key* benötigt. Bei diesem Verfahren kann ein Schlüssel gleichzeitig für die Verschlüsselung mittels AES und für die MIC-Berechnung verwendet werden.

4.3.3.3 AES - Advanced Encryption Standard

Der AES (Advanced Encryption Standard) ist ein symmetrisches Kryptosystem (Blockchiffre), das als Nachfolger für DES (Data Encryption Standard) bzw. 3DES (Triple DES) im Oktober 2000 vom NIST (National Institute of Standards and Technology) der USA als Standard bekannt gegeben wurde. Nach seinen Entwicklern Joan Daemen und Vincent Rijmen wird er auch Rijndael-Algorithmus genannt.

Der Algorithmus besitzt eine variable Blockgröße von 128, 192 oder 256 Bit und eine variable Schlüssellänge von 128, 192 oder 256 Bit. Dieser Algorithmus bietet nach momentanem Kenntnisstand ein sehr hohes Maß an Sicherheit. Das Verfahren wurde eingehenden kryptoanalytischen Prüfungen unterzogen. Der Algorithmus ist frei verfügbar, unterliegt keinerlei Patenten und darf somit ohne Lizenzgebühren eingesetzt und in Soft- oder Hardware implementiert werden.

Der zugrunde liegende Algorithmus kann einfach in Hardware implementiert werden und wurde auch deshalb für WPA2 ausgewählt.

Während dem AES-Algorithmus verschiedene Block- und auch Schlüssellängen zur Auswahl stehen, wurde im IEEE 802.11i-Standard beides jeweils auf 128 Bit festgelegt.

Für weiterführende Informationen zur Implementierung des AES sei auf die Originalpublikationen von Daemen und Rijmen [12] und NIST [13] verwiesen.

4.3.3.4 AES-CCM - AES-CTR/CBC-MAC

Für den IEEE 802.11i Standard wurde der AES-CTR (Counter)-Mode ausgewählt. Zusammen mit dem CBC-MAC (Cipher Block Chaining - Message Authentication Code), der für die Integritätssicherung zuständig ist, bildet dieser den CCM (CTR/CBC-MAC)-Mode und damit die Grundlage für das im IEEE 802.11i Standard verwendete CCMP Verfahren.

CBC-MAC Um den MIC zu berechnen, bedarf es im CCMP keines zusätzlichen Verfahrens wie beim TKIP. AES-CBC-MAC verwendet dazu den folgenden Ablauf:

1. Ein 128 Bit Startblock der Daten wird mit AES und dem Datenintegritätsschlüssel verschlüsselt. So entsteht ein 128 Bit langes Ergebnis (E1).
2. Es wird ein exklusives Oder (XOR) mit E1 und den nächsten 128 Bit der Daten, für die die MIC berechnet wird, durchgeführt. Es entsteht ein weiteres 128 Bit langes Ergebnis (XE1).
3. XE1 wird mit AES und dem Datenintegritätsschlüssel verschlüsselt. Dies ergibt E2.
4. Es wird ein XOR mit E2 und den nächsten 128 Bit der Daten durchgeführt. Daraus resultiert X2.

Die Schritte 3. und 4. werden für alle weiteren 128 Bit Blöcke wiederholt. Die höchsten 64 Bit des letzten 128 Bit Ergebnisses ist der WPA2 MIC-Wert.

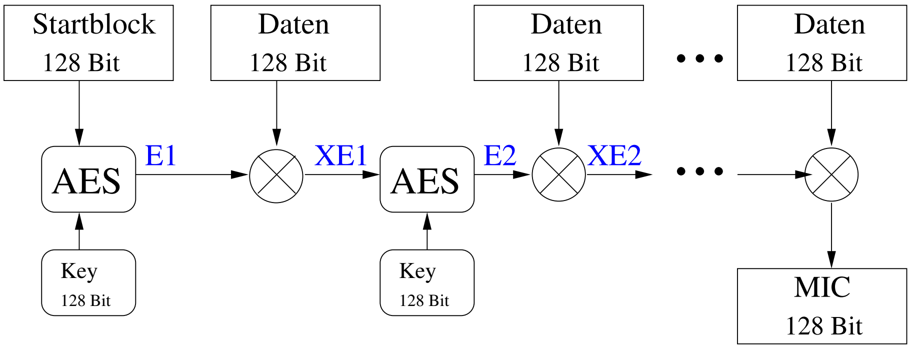

Abbildung 4.7: Schematische Darstellung von CBC-MAC

AES-CTR Der AES-CTR (AES-Counter-Mode) enthält einen zusätzlichen Zähler. AES wird nicht auf den zu verschlüsselnden Datenblock angewendet, sondern zusammen mit dem Schlüssel für die Datenverschlüsselung auf den Zähler. Das Ergebnis dieser Verschlüsselung wird dann mit dem eigentlichen Teil der Nachricht mittels XOR verknüpft. Dieser 128 Bit große Zähler setzt sich dabei aus zwei 1 Byte großen *Flag*- bzw. *Prio*-Feldern, der MAC-Adresse des Senders (6 Byte), der Paketnummer (6 Byte) sowie einem 2 Byte großem *Counter*-Feld zusammen. Dieses Feld wird auf 1 gesetzt und je Block um 1 erhöht. Damit wäre es theoretisch möglich, 2^{16} 128 Bit Blöcke innerhalb eines WLAN-Pakets zu verschlüsseln, ohne dass sich der Zähler wiederholt.

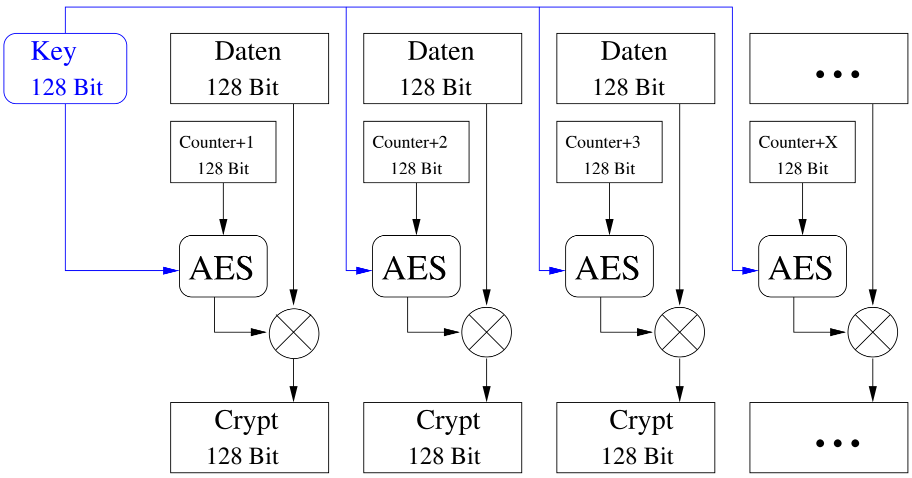

Abbildung 4.8: Schematische Darstellung von AES-CTR

4.3.3.5 CCMP-Verschlüsselungsablauf

Um ein Unicast-Datenpaket zu verschlüsseln, nutzt CCMP das folgende Verfahren:

1. Startblock, 802.11-MAC-Header, CCMP-Header, Daten-Feld und Padding-Felder werden dem AES-CBC-MAC-Algorithmus übergeben. Zusammen mit dem Datenintegritätsschlüssel wird der MIC erstellt.
2. Start-Counter und kombinierte Daten mit der MIC werden dem AES-CTR-Algorithmus übergeben.
3. Es wird ein CCMP-Header mit der Paketnummer zum verschlüsselten Teil der 802.11-Daten hinzugefügt. Das Ergebnis wird in einen 802.11-Header und Trailer gekapselt.

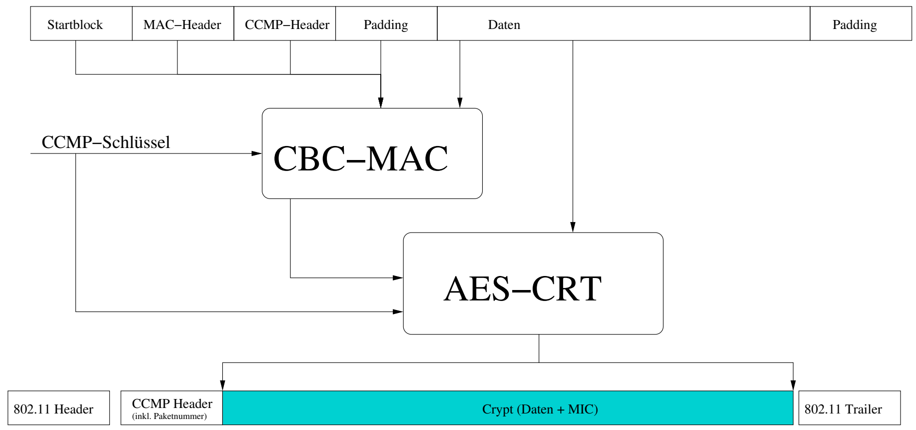

Abbildung 4.9: CCMP-Datenframe-Verschlüsselung

4.3.3.6 CCMP-Entschlüsselungsablauf

Um ein Unicast-Datenpaket zu entschlüsseln und die Datenintegrität zu überprüfen, nutzt CCMP das folgende Verfahren:

1. Der Start-Counter wird über die 802.11- und CCMP-Header ermittelt.
2. Der Start-Counter und der verschlüsselte Teil der 802.11-Daten werden mit dem Schlüssel für die Datenverschlüsselung an den AES-CTR-Algorithmus übergeben. Dieser erstellt die unverschlüsselten Daten und die MIC.
3. Startblock, 802.11-MAC-Header, CCMP-Header, Daten-Feld und Padding-Felder werden dem AES-CBC-MAC-Algorithmus übergeben, der mit dem Datenintegritätsschlüssel eine MIC berechnet.

4. Der berechnete MIC und der entschlüsselte MIC werden verglichen. Wenn sie nicht übereinstimmen, werden die Daten vom CCMP verworfen.

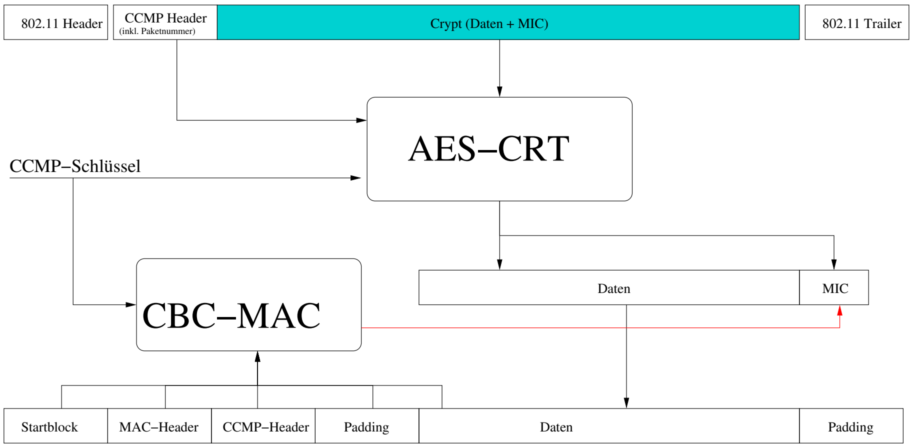

Abbildung 4.10: CCMP-Datenframe-Entschlüsselung

4.3.4 Hardware-Adressen-Filter

Ebenfalls auf der Sicherungsschicht (ISO-OSI Layer 2) arbeitet der Hardware-Adressen-Filter. Er gehört zu den *Access Control Lists* und filtert anhand von Listen mit den Hardwareadressen bestimmter Netzwerkkarten. Der Access Point lässt dann entweder nur die auf der Liste stehenden Einträge zu (engl. Whitelist) oder schließt sie vom Netzzugang aus (engl. Blacklist).

Die Hardware-Adressen-Filter stellt weder Verschlüsselung noch Integritätsprüfung bereit. Sie dient lediglich als einfaches Verfahren für die Zugangskontrolle. Allerdings lassen sich Absender-MAC-Adressen leicht fälschen und somit bietet das Verfahren keinerlei Schutz. Es wurde hier nur der Vollständigkeit halber mit aufgelistet.

4.4 VPN - Virtual Private Network

Nachdem im Abschnitt 4.3 Verschlüsselungsverfahren auf der Sicherungsschicht des ISO-OSI-Modells genauer untersucht wurden, soll es in diesem Kapitel um Datenverschlüsselungen auf den darüberliegenden Schichten gehen. Man spricht in dem Zusammenhang oft von VPN (engl. Virtual Private Network).

Ursprünglich war der Haupteinsatzzweck sogenannter VPNs die Bereitstellung einer gesicherten Kommunikation über öffentliche Datenleitungen (Internet), um z. B. Firmenfilialen miteinander zu verbinden oder Außendienstmitarbeitern einen Zugang zum Firmennetzwerk zu ermöglichen. Damit ist es möglich, Kosten für separat gemietete Leitungen zu sparen. Dazu simuliert das VPN mittels sogenannter *Tunnel* die Übertragung des Datenverkehrs auf einer isolierten Datenleitung. Tatsächlich wird der Datenverkehr über die öffentlichen ungeschützten Datenleitungen (z. B. über WAN (Wide Area Network)) übertragen.

Vom Aufbau ähnelt das WLAN eher der Struktur eines Intranets, doch aufgrund der unzähligen Sicherheitslücken, die bis zur Einführung des 802.11i-Standards im letzten Jahr bestanden, ist es eher mit dem Internet zu vergleichen. Es ist daher damit zu rechnen, dass Daten abgehört oder verfälscht werden können.

So lag es also nahe, die bereits seit den 90er Jahren bewährte Sicherheitstechnologie der VPNs auch auf das WLAN anzuwenden. Um diese Technologie damit zu nutzen, bedurfte es keiner großen Anstrengung, da sich die bestehenden Techniken problemlos darauf abbilden lassen.

Unter einem *Tunnel* im Sinne von VPNs ist ein Verfahren zu verstehen, nach dem ein zu sendendes Datenpaket komplett in ein weiteres Paket gekapselt (im Allgemeinen auch verschlüsselt) wird, um dann als neues Datenpaket über das Netz versendet zu werden. Auf der Empfängerseite wird dann das eigentliche Datenpaket aus dem empfangenen Paket wieder extrahiert (und entschlüsselt). Der Transport der Daten erfolgt damit wie durch einen Tunnel. Für eine sinnvolle Sicherung des *Tunnels* ist es notwendig, dass sich Sender und Empfänger vor der Etablierung der *Tunnel*-Verbindung authentifizieren. Neben der Authentifizierung ist auch eine Verschlüsselung der Daten sinnvoll.

Für den Einsatz eines Tunnels wird ein Trägerprotokoll benötigt, mit dem die gekapselten Nachrichten transportiert werden und ein Verschlüsselungsprotokoll, mit dem die Originalnachricht vor unbefugtem Ausspähen geschützt wird.

Im Prinzip wären hier beliebige Kombinationen von Protokollen möglich, durchgesetzt haben sich allerdings nur einige wenige, die hier in den folgenden Abschnitten kurz vorgestellt werden sollen.

4.4.1 IPSec - IP Security

IPSec (IP Security) wurde 1998 entwickelt, um die Mängel bzw. Schwächen von IP zu beheben. Wie der Name bereits vermuten lässt, stellt es eine Architektur für die sichere Kommunikation im IP-Netz dar. Somit werden Vertraulichkeit, Authentizität und Integrität gewährleistet.

Dabei ist IPSec sowohl Trägerprotokoll als auch für die Verschlüsselung zuständig. Dies ist deshalb möglich, weil IPSec aus einer Reihe verschiedener Protokollnormen besteht, die auch miteinander kombiniert werden können.

IPSec bietet aber noch mehr als nur einen verschlüsselten Übertragungsmechanismus in Form des ESP (Encapsulating Security Payload; RFC 2406 [14]). Es ermöglicht darüber hinaus, die Integrität und die Authentizität der Nachricht zu garantieren (mittels AH (Authentication Header; RFC-2402 [15])) und stellt sogar Mittel für ein integriertes Schlüsselmanagement (mittels IKE (Internet Key Exchange; RFC-2409 [16])) zur Verfügung. Alle Merkmale von IPSec werden in den RFCs 2401 bis 2409 beschrieben.

IPSec kennt zwei Betriebsmodi: Transportmodus und Tunnelmodus.

Transportmodus Im Transportmodus wird ausschließlich der Datenteil (Nutzdaten, „Payload”) des IP-Pakets verschlüsselt. Alle anderen Teile der Nachricht (also der IP-Header) bleiben unverändert. Den Nutzdaten wird nur der ESP-Header vorangestellt.

IP	ESP	IP	TCP	DATEN

verschlüsselter Bereich

Abbildung 4.11: IPSec Transport-Mode

Tunnelmodus Im Tunnelmodus wird das komplette IP-Paket vor der Übertragung verschlüsselt und mit einem neuen IP-Header versehen, der die Daten für das Ziel-Gateway enthält. Auf diese Weise können transparente Verbindungen, z. B. zwischen zwei Firmenstandorten, über das Internet hergestellt oder ein einzelner Rechner (z. B. Notebook) über ein öffentliches (IP-)Netzwerk mit einem Firmennetz verbunden werden.

IP	ESP	TCP	DATEN

verschlüsselter Bereich

Abbildung 4.12: IPSec Tunnel-Mode

4.4.1.1 Architektur

Die zentralen Funktionen in der IPSec-Architektur sind das AH-Protokoll, das ESP-Protokoll und die Schlüsselverwaltung (IKE).

Echtheit (AH) Das *AH*-Protokoll sorgt für die Echtheit der zu übertragenen Daten und Protokollinformationen. *AH* wird zum Schutz der Paketintegrität und der Paketechtheit eingesetzt. Dies erfolgt über ein Hashing-Verfahren (MD5 und SHA). Da der äußere IP-Header in die Hashprüfung mit einbezogen wird, ist auf *AH*-Pakete keine Art von NAT (Network Address Translation) anwendbar.

Im *AH*-Paket befindet sich neben verschiedenen anderen Feldern der Security Profile Index (SPI) und eine Sequenznummer. Diese Sequenznummer kann zum Schutz vor Replay-Angriffen verwendet werden.

AH nutzt die IP-Protokoll-Nummer 51. Das *AH*-Protokoll kann sowohl im Transport-, als auch im Tunnelmodus eingesetzt werden.

Verschlüsselung (ESP) Das *ESP*-Protokoll dient zur Verschlüsselung eines Datenpaketes und, mittels Hashing-Verfahren, auch der Integritätssicherung. Zur Verschlüsselung können beliebige Verfahren eingesetzt werden. Allerdings muss DES unterstützt werden (in der Praxis wird meistens auch 3DES implementiert). Als Hashing-Verfahren müssen *MD5-96* (RFC-2403 [17]) und *SHA-96* (RFC-2404 [18]) verfügbar sein.

Im *ESP*-Header befinden sich, wie bereits beim *AH*-Paket, der *Security Profile Index* (SPI) und eine Sequenznummer. *ESP* nutzt die IP-Protokoll-Nummer 50.

Schlüsselverwaltung (IKE) Für die Schlüsselverwaltung gibt es zwei Möglichkeiten die Verwaltung und Verteilung der Schlüssel durchzuführen. Neben der reinen, heute nicht mehr sehr gebräuchlichen, manuellen Schlüsselverwaltung, kann das *IKE* eingesetzt werden. Um eine gesicherte Verbindung zwischen zwei Stationen aufzubauen, müssen auf beiden Seiten viele Parameter ausgetauscht werden:

- Art der gesicherten Übertragung (Echtheit oder Verschlüsselung)
- Verschlüsselungsalgorithmus
- Schlüssel
- Dauer der Gültigkeit der Schlüssel
- weitere Werte

Alle diese Parameter werden in einer SA (Security Association) beschrieben. Jede gesicherte Verbindung bedarf einer solcher *SA* an jedem Ende einer logischen Verbindung.

Die Hauptmethode des *IKE* unterstützt keine dynamischen IP-Adressen und keine Art von NAT. Für solche Fälle muss *IKE Aggressive Methode* eingesetzt werden, bei der ein Angreifer den Initiator einer Verbindung feststellen kann.

IKE nutzt den UDP-Port 500.

4.4.1.2 Zusammenfassung

Ausführliche Informationen zum IPSec sind im Teil 3 von Jon C. Snader „*VPNs Illustrated: Tunnels, VPNs, and IPSec*" [19] oder in „*IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks, Second*" [20] zu finden.

IPSec ist in aktuellen Betriebssystemen bereits integriert und lässt sich somit ohne zusätzliche Software nutzen. Allerdings ist die Konfiguration recht komplex und damit fehleranfällig. Ein großer Nachteil ist außerdem, dass IPSec nicht NAT-fähig ist (siehe oben). Weiterhin stellt IPSec nur eine sichere Verbindung zwischen Geräten, jedoch nicht zwischen Benutzern her. Daher ist eine weitere Authentifizierungsinstanz notwendig.

Eine weitere Einschränkung von IPSec besteht darin, dass es auf das Tunneln von IP-Paketen beschränkt ist. Dienste, die auf andere Protokolle wie beispielsweise das X.25-Protokoll oder IPX (Internet Packet Exchange) von Novell angewiesen sind, können damit nicht übermittelt werden.

4.4.2 PPTP - Point-to-Point Tunneling Protocol

Das PPTP (Point-to-Point Tunneling Protocol) ist ein von verschiedenen Herstellern (darunter Microsoft) entwickeltes Protokoll zum Aufbau eines VPN. Hierbei werden *PPP* (Point-to-Point-Protocol)-Pakete [21] durch ein IP-Netzwerk getunnelt. Die *PPP*-Pakete sind wiederum in GRE (Generic Routing Encapsulation Protocol; RFC 1701 [22] und RFC 1702 [23]) gekapselt.

PPTP ermöglicht nicht nur die Übertragung von IP-Paketen, sondern auch von IPX-Paketen. Deshalb spricht man hier auch von einem *Layer-2-Tunneling-Protokoll*.

Zur Authentifizierung dient PAP [24] oder CHAP [25]. In PPTP ist kein Schlüsselmanagement spezifiziert. Die Datenverschlüsselung erfolgt nach dem RC4-Verfahren mit Schlüssellängen von 40, 56 oder 128 Bit. Eine Paket-Integritätsprüfung ist ebenfalls nicht spezifiziert.

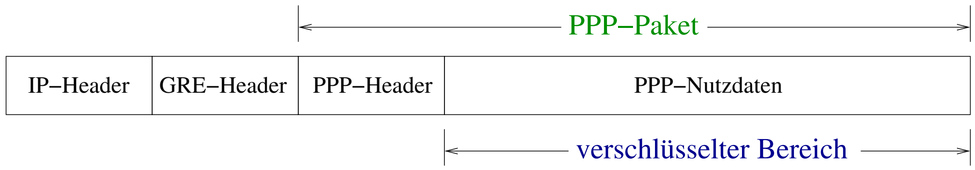

Abbildung 4.13: PPTP-Paket

PPTP war bei der Implementierung nicht standardisiert. Der PPTP-Standard wurde erst später durch ein informelles RFC (RFC 2637 [26]) im Jahre 1999 veröffentlicht.

4.4.3 L2TP - Layer 2 Tunneling Protocol

L2TP (Layer 2 Tunneling Protocol) stammt von Cisco und wurde in der RFC 2661 [27] spezifiziert. Es stellt eine Kombination von PPTP und L2F (Cisco Layer Two Forwarding (Protocol); RFC 2341 [28]) dar und vereinigt deren Vorteile. So ist es durch die Tunnel-ID im L2TP-Header möglich, anders als beim PPTP, mehrere *Tunnel* parallel zu betreiben. Außerdem wird NAT unterstützt. Weiterhin funktioniert L2TP auch über X.25, Frame Relay und ATM (Asynchronous Transfer Mode).

L2TP erlaubt eine Authentifizierung auf der Basis von PAP oder CHAP. Es ist keine Verschlüsselung definiert.

L2TP über IP verwendet UDP und eine Reihe von L2TP Nachrichten für die Verwaltungsaufgaben des *Tunnel*. L2TP verwendet zudem UDP um L2TP-gekapselte PPP-Pakete als getunnelte Daten zu versenden.

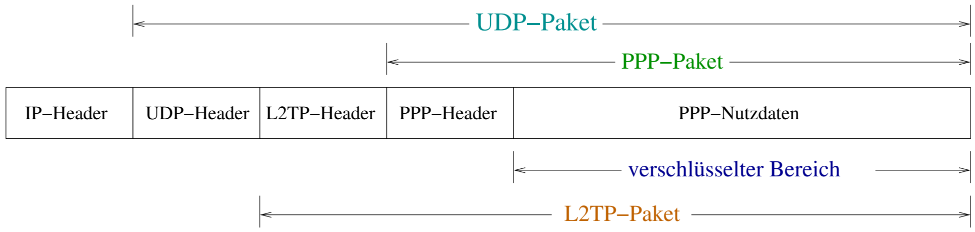

Abbildung 4.14: L2TP-Paket

In der RFC 3193 [29] ist eine Methode beschrieben, um L2TP und IPSec zu kombinieren und L2TP dadurch mit der fehlenden Verschlüsselung zu versehen.

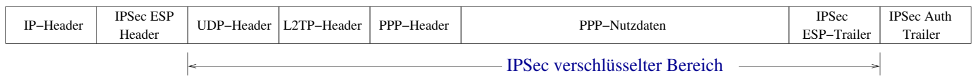

Abbildung 4.15: L2TP-Paket mit IPSec

4.4.4 SSL/TLS - Secure Socket Layer/Transport Layer Security

SSL/TLS ist kein VPN im zuvor genannten Sinne. Es dient viel mehr der Absicherung von Applikationsprotokollen, wie z. B. HTTP, POP3, IMAP, SMTP usw. Es ist oberhalb des ISO-OSI Layer 4 (Transportschicht) und unterhalb des ISO-OSI Layer 7 (Anwendungsschicht) angesiedelt. SSL/TLS arbeitet transparent und kann somit leicht eingesetzt werden, um Protokollen ohne eigene Sicherungsmechanismen eine gesicherte Verbindung zur Verfügung zu stellen.

SSL wurde von Netscape entwickelt. Die erste Version (SSL 1.0) erschien 1994. Danach erschienen noch die Versionen SSL 2.0 und 3.0 bevor es von der IETF 1999 in der RFC 2246 [30] spezifiziert wurde.

SSL/TLS ist als ein zweischichtiges Protokoll definiert. Auf der unteren Ebene ist das *SSL-Record-Protocol*, das die übergebenen Daten in Pakete zufälliger Größe fragmentiert, komprimiert und mit einem *Message Authentification Code* versieht.

Hier die wichtigsten symmetrischen Verschlüsselungsverfahren mit den zugehörigen *Message Authentification Codes*: DES (SHA-1), 3DES (SHA-1), RC-4 (MD5)

Auf das *SSL Record Protocol* setzen das *Handshake-*, das *Change-Cipher-Spec-* und *Alert*-Protokoll auf. Das *Handshake*-Protokoll legt den Ablauf der Authentifizierung von Server und Client fest. Das *Change-Cipher-Spec-Protokoll* regelt den Wechsel einer neu ausgehandelten *Cipher-Suite* und das *Alert*-Protokoll dient zum Austausch von Kontrollnachrichten. Die *Cipher-Suite* enthält den verwendeten symmetrischen Verschlüsselungsalgorithmus, den Schlüsselaustauschalgorithmus sowie die Hash-Funktion.

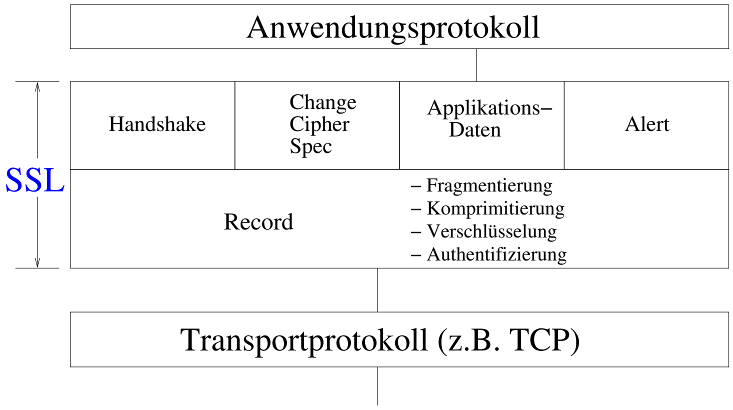

Abbildung 4.16: Elemente des SSL/TLS

In letzter Zeit wird das SSL/TLS Protokoll auch immer öfter zum Aufbau von VPN-Tunneln benutzt. Dabei muss man zwei Einsatzgebiete unterscheiden: Anwendungen, die sich über den Webbrowser bedienen lassen und die, die sich damit nicht bedienen lassen.

Im ersten Fall greift der Anwender mit seinem Browser auf einen Webserver, der gleichzeitig das Tunnelende im Heim-Netz darstellt, zu. Im zweiten Fall müssen die Daten per *Active-X-Applet* oder *Java-Applet* in ein „browser-konformes” Format gebracht werden, d. h. das Applet emuliert den eigentlichen Client.

Der Vorteil dieser Lösung besteht darin, dass der Zugriff über dieses VPN ins Firmennetz ohne eigenen VPN-Client vorgenommen werden kann, also z. B. auch aus einem Internet-Café. Für den zweiten Fall müssen jedoch die benötigten Applets installiert werden.

Der Nachteil besteht darin, dass es sich bei SSL eigentlich um kein *Tunnel*-Protokoll im oben genannten Sinne handelt. Die Daten müssen daher immer auf die Anwendungsschicht des ISO-OSI-Modelles durchgereicht und gegebenenfalls emuliert werden, was sich negativ auf die Performanz auswirkt. Zumindest im zweiten Fall besteht zusätzlich das Problem der korrekten Arbeitsweise des „Active-X/Java-Emulators”.

4.4.5 Lightweight VPN

Weiterhin gibt es noch eine Menge weiterer Möglichkeiten ein VPN aufzubauen. Hier wären die sogenannten *Lightweight VPN* zu nennen. Als *lightweight* (leichtgewichtig) werden sie bezeichnet, weil sie relativ einfach in ihrer Anwendung sind. In der Regel wird lediglich eine Zusatzsoftware installiert und ohne großem Konfigurationsaufwand kann eine gesicherte Verbindung aufgebaut werden. Der Nachteil ist, dass sie oft nur eine beschränkte Betriebssystemunterstützung haben.

In diesen Bereich fallen z. B. VTun [31], CIPE [32], oder OpenVPN [33].

5 EAP und 802.1x

5.1 EAP - Extensible Authentication Protocol

Die IETF (Internet Engineering Task Force) standardisierte EAP ursprünglich für das PPP-Protokoll. Um zu verhindern, dass viele verschiedene Authentifizierungsverfahren für PPP separat entwickelt werden und damit eine Unmenge von Protokollnummern verschwendet werden, entschied sich die IETF für die Entwicklung eines einheitlichen Authentifizierungsverfahrens zur Kapselung von beliebigen Authentifizierungsprotokollen. EAP arbeitet dabei auf der Sicherungsschicht (ISO-OSI Layer 2) und wurde Anfang 1998 durch die RFC 2284 [34] standardisiert.

5.1.1 EAP-Paket-Format

Ein EAP-Paket besteht aus vier Feldern. Abbildung 5.1 zeigt den generellen Aufbau eines EAP-Pakets und Tabelle 5.1 beschreibt die einzelnen Felder näher.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Data ...
+-+-+-+-+
```

Abbildung 5.1: Aufbau: EAP-Paket

Type	Beschreibung
Code	Dieses Feld ist 1 Byte groß und identifiziert den Typ des EAP-Pakets. Es gibt somit an, wie das Datenfeld zu interpretieren ist.
Identifier	Dieses Feld ist 1 Byte groß um zusammengehörende Requests und Responses zu identifizieren. Bei wiederholten Übertragungen bleibt der Identifikator gleich, bei neuen Übertragungen wird ein neuer Identifikator verwendet.
Length	Dieses Feld ist 2 Byte groß und enthält die Gesamtlänge des Pakets (in Byte)
Data	Das Datenfeld ist von variabler Länge. Die Interpretation der Daten ist abhängig vom Wert des Code-Feldes.

Tabelle 5.1: EAP-Paket-Format-Felder

5.1.2 EAP-Request/Response

Ein EAP-Datenaustausch besteht aus *Requests* und *Responses*. Der Authentifikator sendet einen Identifizierungs-*Request* an das System, das sich für den Netzwerkzugang anmelden will. In Abhängigkeit eines gültigen Identifizierungs-*Response* gewährt der Authentifikator den Zugang zum Netzwerk, anderenfalls wird der Zugang verweigert.

Abbildung 5.2 zeigt den Aufbau der EAP-Request- und EAP-Response-Pakete. Sie folgen dabei dem Aufbau der allgemeinen EAP-Pakete (siehe Abbildung 5.1) und teilen das Datenfeld dabei in ein 1 Byte großes Datentypfeld und einer zum Datentyp passendes Feld variable Größe.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Type      |  Type-Data ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Abbildung 5.2: Aufbau: EAP-Request/Response

Type

In jedem Paket wird genau ein Typ angegeben. Dabei haben zusammengehörige Requests und Responses den gleichen Typ (Ausnahme: NAK (New Authentication Method)). Typbezeichner größer *3* weisen auf die Authentifizierungsmethode hin.

Type-Data

Das Datenfeld ist von variabler Größe. Der Inhalt des Feldes wird entsprechend des Typs aus dem *Type*-Feld interpretiert.

Tabelle 5.2 gibt einen Überblick der verschiedenen Typen und deren Bedeutung.

Code	Typ	Beschreibung
1	Identity	Das initiale Paket vom Authentifikator an das Benutzersystem. Das Datenfeld enthält z. B. den Text, den der Benutzer als Eingabeaufforderung angezeigt bekommt.
2	Notification	Der Authentifikator kann den *Notification*-Typ benutzen, um dem Benutzer Nachrichten zu senden, z. B. über ein bald ablaufendes Passwort.
3	NAK	NAK wird vom Benutzersystem verwendet um eine neue Authentifizierungsmethode dem Authentifikator vorzuschlagen, weil es die vorgeschlagene Methode nicht benutzen will oder nicht unterstützt.

Fortsetzung siehe nächste Seite

4	MD5 Challenge	Das MD5-Verfahren muss von allen Implementationen unterstützt werden. Es kann aber durch ein NAK einer anderen Methode der Vorzug gegeben werden.
5	OTP (One Time Password)	Ein Verfahren mit nur einmalig gültigen Passworten, um Abhörangriffen entgegenzuwirken. Nachteil ist, dass hierzu eine Infrastruktur existieren muss, die diese Passworte verwaltet.
6	Generic Token Card	Diese Methode nutzt das RSA SecurID-Verfahren, das zufällige Passwörter in bestimmten Zeitintervallen erzeugt, um wie bei OTP nur einmal gültige Passworte zu generieren.
13	TLS	Das TLS-Verfahren ist zur gegenseitigen Authentifizierung geeignet, garantiert sicheren Schlüsselaustausch und ist außerdem zur Erzeugung sicherer Schlüssel geeignet. Es benötigt eine PKI und eine CA (Certificate Authority), die die Authentizität der verwendeten Zertifikate garantiert.
17	LEAP	Cisco Light EAP basiert auf der Authentifizierung mittels Login und Passwort. Es handelt sich hierbei um ein von Cisco entwickeltes proprietäres Protokoll und unterstützt Session-Keys, die nach einer bestimmten Zeit getauscht werden.
21	TTLS	Tunneled Transport Layer Security basiert auf TLS, benötigt jedoch keine PKI. Der *Tunnel* wird mittels Serverzertifikat aufgebaut und kann sowohl weitere EAP- oder andere Authentifizierungsmethoden (PAP, CHAP usw.) transportieren.
25	PEAP	Protected EAP funktioniert ähnlich wie TTLS und setzt dabei ebenfalls auf die *Tunnel*-Funktion. Allerdings unterstützt PEAP nur den Transport anderer EAP-Authentifizierungen. Es ist proprietär und wird hauptsächlich durch Microsoft und Cisco unterstützt.

Tabelle 5.2: EAP-Request/Response-Type-Codes

5.1.3 EAP-Success/Failure

Für den Abschluss einer Authentifizierung sendet der Authentifikator an das Benutzersystem ein EAP-Paket mit dem Type-Code *3* für die erfolgreiche oder den Type-Code *4* für fehlgeschlagene Authentifizierung. Diese Pakete werden deshalb als *EAP-Success* bzw. *EAP-Failure* bezeichnet und haben folgenden Aufbau:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Abbildung 5.3: Aufbau: EAP-Sucess/Failure

5.2 802.1x - Network Port Authentication

Aus verschiedenen Gründen wurde für Netzwerke eine Zugangssteuerung ähnlich des PPP, wie es bei der herkömmlichen Einwahl beim ISP (Internet Service Provider) üblich ist, notwendig. So entwickelte die IEEE den Standard 802.1x, der eine Authentifizierung für den Netzwerkzugang, ähnlich der PPP Authentifizierung, darstellt.

Dabei setzt 802.1x auf die Sicherungsschicht des ISO-OSI Modells auf und verwendet als Authentifizierungsverfahren EAP.

5.2.1 802.1x Architektur

Zur Authentifizierung werden in 802.1x drei Komponenten definiert. Zum einen das Benutzersystem, auch als Supplikant bezeichnet. Dem gegenüber stehen der Authentifikator (engl. Authenticator) und der Authentifizierungsserver (engl. Authentication Server). Benutzersystem und Authentifikator werden im Standard auch PAE (Port Authentication Entities) genannt. Abbildung 5.4 verdeutlicht die Beziehungen der Komponenten zueinander.

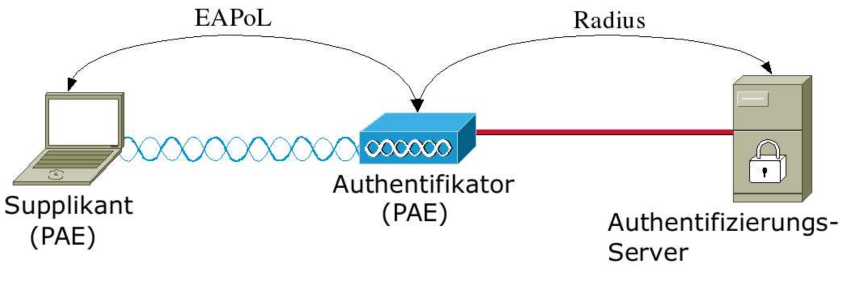

Abbildung 5.4: 802.1x Architektur

Der Authentifikator arbeitet als Mittelsmann und hat keine Kenntnis von Benutzerdaten, sondern reicht die Zugangsanfragen vom Benutzersystem an den Authentifizierungs-Server weiter. Dazu wird zwischen dem Benutzersystem und dem Authentifikator das EAP-Protokoll verwendet. Zwischen Authentifikator und Authentifizierungsserver wird ein entsprechendes AAA-Protokoll (siehe Kapitel 6, z. B. RADIUS) verwendet. 802.1x ist also eine Spezifikation, die die Zusammenarbeit dieser verschiedenen Verfahren koordiniert. Es beschreibt die Mechanismen für Anfragen zur Zugangsberechtigung sowie die Kommunikation mit der Beglaubigungsinstanz.

Hierbei steuert der Authentifikator den Zugang des Benutzersystems zum Netzwerk. Dazu können sich die Anschlüsse des Gerätes in zwei Zuständen befinden:

- **unauthorized** - Der Authentifikator lässt nur die für die Authentifizierung notwendigen Daten passieren.
- **authorized** - Die Authentifizierung war erfolgreich, das Gerät darf am Netzwerk teilnehmen.

Die Authentifizierung ist im Authentifizierungsserver implementiert und kann dort bei Bedarf ausgewechselt werden, ohne am übrigen Ablauf etwas zu ändern.

5.3 EAPoL - Extensible Authentication Protocol over LAN

Bei drahtgebundenen Netzwerken mit physischen Ports (z. B. Port am Switch) kommt bei 802.1x EAPoL zum Einsatz. Die Besonderheit bei EAPoL sind die *Start*- und *Logoff*-Frames, die zur Anmeldungsinitialisierung bzw. für die Abmeldung dienen.

5.3.1 Encapsulation

Da EAP auf der Sicherungsschicht (ISO-OSI Layer 2) arbeitet, werden diese Daten in einen Ethernet-Rahmen verpackt. Abbildung 5.5 zeigt ein komplettes EAPoL-Paket. Die einzelnen Felder werden in Tabelle 5.3 näher beschrieben.

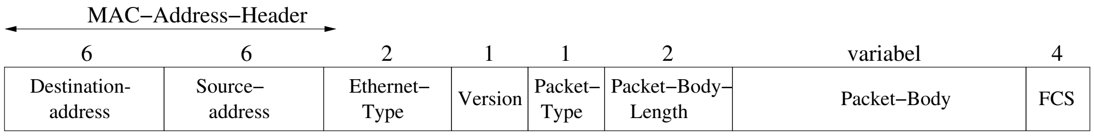

Abbildung 5.5: Aufbau eines EAPoL-Frames

Feld	Bytes	Beschreibung
MAC Header	12	Quell- und Ziel-MAC-Adresse
Ethernet Type	2	Das Feld enthält den Code für EAPoL: 88-8e.
Version	1	Bisher ist nur Version 1 standardisiert.
Packet Type	1	EAPoL-Erweiterung um EAP das portbasierte LAN anzupassen (siehe Tabelle 9.1).
Packet Body Length	2	Die Länge des *Packet Body* (0, wenn es keinen Body gibt).
Packet Body	x	Ein Feld variabler Länge. Es entfällt bei den Packet Types EAPoL-Start und EAPoL-Logoff. Im Fall von EAP-Packet, EAPoL-Key oder EAPoL-Encapsulated-ASF-Alert kapselt es ein EAP-Paket, einen Key oder einen Alert.
FCS	4	Frame Check Sequence.

Tabelle 5.3: Beschreibung der EAPoL-Felder

Ein Überblick der verschiedenen EAP-Paket-Typen ist im Anhang (Tabelle 9.1) zu finden.

EAPoL-Nachrichten werden im LAN von Benutzersystemen an die vom IEEE vergebene Multicast-Adresse 01:80:c2:00:00:03 gesendet, solange keine Hardware-Adresse des Gegenübers bekannt ist.

5.3.2 EAPoL-Ablauf

Der EAPoL-Ablauf ähnelt stark dem allgemeinen EAP-Ablauf, wobei zusätzlich Gebrauch vom *EAPoL-Start* und *EAPoL-Logoff* gemacht wird.

Abbildung 5.6 zeigt den Ablauf einer vollständigen Authentifizierung mittels EAPoL. Der Authentifizierungsserver ist in diesem Beispiel ein RADIUS-Server eingesetzt. Deshalb „übersetzt” der Authentifikator in diesem Fall vom EAPoL in das RADIUS-Protokoll (siehe Abschnitt 6.2.2).

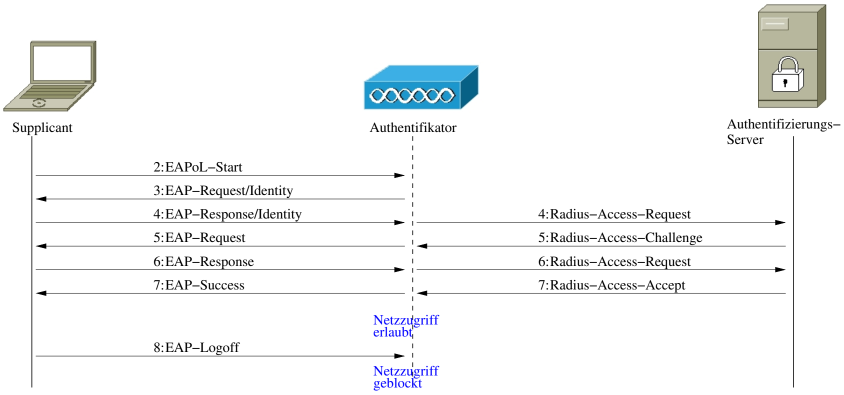

Abbildung 5.6: EAPoL Ablaufschritte

Ablaufbeschreibung:

1. Der Port zum Benutzersystem ist im *unauthorized* Zustand, d. h. der Netzwerkzugang wird verwehrt.

2. Der Supplikant startet den Austausch mit einer *EAPoL-Start*-Nachricht.

3. Der „normale” EAP-Austausch beginnt, indem der Authentifikator ein EAP-Request/Identity-Paket schickt.

4. Darauf antwortet das Benutzersystem mit einem *EAP-Response/Identity*, das vom Authentifikator als *RADIUS-Access-Request* weitergeleitet wird.

5. Der RADIUS-Server antwortet mit einem *RADIUS-Access-Challenge*-Paket, das vom Authentifikator unter Auswahl des geeigneten Protokolls mit allen notwendigen Daten an das Benutzersystem übermittelt wird.

6. Dieser wiederum sendet die vom Benutzer eingegebenen Daten als *EAP-Response* zurück an den Authentifikator, der daraufhin die Ergebnisdaten in das Daten-Feld eines *RADIUS-Access-Requests* verpackt und weiterleitet.

7. Der RADIUS-Server bewilligt den Zugriff mit einem *RADIUS-Access-Accept*, woraufhin der Authentifikator ein *EAP-Success* an das Benutzersystem versendet und den Port in den *authorized*

Zustand schaltet. Der Benutzer ist für die Benutzung des Netzwerkes authentifziert und kann mit seinem System auf das Netz zugreifen.

8. Um den Zugriff zum Netzwerk zu beenden wird vom Benutzersystem ein *EAPoL-Logoff* an den Authentifikator gesendet, der wiederum den Port in den *unauthorized* Zustand zurücksetzt.

Anmerkung: Das Benutzersystem muss nicht zwangsweise eine *EAPoL-Start*-Nachricht senden. Der Authentifikator kann jederzeit ein *EAP-Request/Identity* versenden, um die Authentifizierungsdaten auf den neuesten Stand zu bringen.

5.4 EAPoW - Extensible Authentication Protocol over WLAN

Auch für das WLAN lässt sich das 802.1x anwenden. Dazu muss der Begriff „Network-Port" aus der 802.1x Spezifikation auf die WLAN-Technologie übertragen werden. Nach IEEE ist die *Association* zwischen Station und Access Point als „logischer Port" (engl. logical port) zu verstehen. Er existiert allerdings erst dann, wenn die *Association* erfolgreich abgeschlossen ist.

Wenn also eine Verbindung nach 802.11 aufgebaut ist, verwirft der Access Point sämtlichen Datenverkehr, bis auf die 802.1x Daten, solange die Authentifizierung nicht erfolgreich abgeschlossen ist.

Dabei kann EAPoW auf die Nutzung der oben genannten Multicast-Adresse verzichten, weil bei EAPoW bereits durch die *Association* die Hardware-Adresse der Gegenstelle bekannt ist.

EAPoW bietet die Möglichkeit, mittels *EAP-Key* dynamisch Sitzungsschlüssel zu übertragen.

5.4.1 EAPoW-Ablauf

Der Ablauf von EAPoW ähnelt sehr stark dem Ablauf von EAPoL. Abbildung 5.7 zeigt anhand eines Beispieles den Ablauf von EAPoW. Auch hier kommt wieder das RADIUS-Protokoll zwischen Authentifikator und Authentifizierungsserver zum Einsatz.

Ablaufbeschreibung:

1. Der Wireless-Client sendet einen *Association-Request*, um seinen Verbindungsaufbauwunsch dem Access Point zu signalisieren.

2. Der Access Point lässt dies zu und sendet dazu den *Association-Response* als Antwort. Die Verbindung ist aufgebaut und auf dem Access Point existiert zu dieser Verbindung ein „logischer Port".

3. Der Supplikant startet den Austausch mit einer *EAPoL-Start*-Nachricht.

4. Der Authentifikator (hier der Access Point) schickt einen *EAP-Request/Identity* an den WLAN-Client.

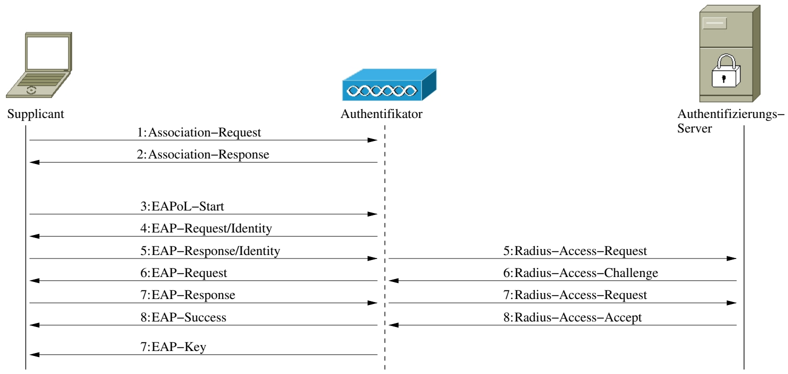

Abbildung 5.7: EAPoW Ablaufschritte

5. Dieser antwortet darauf mit einem *EAP-Reponse/Identity*, der vom Authentifikator als *RADIUS-Access-Request* an den Authentifizierungsserver weitergeleitet wird.

6. Der RADIUS-Server antwortet mit einer *RADIUS-Access-Challenge*-Nachricht, die vom Authentifikator unter Auswahl des geeigneten Protokolls mit allen notwendigen Daten an den WLAN-Client übermittelt wird.

7. Daraufhin werden die vom Benutzer eingegebenen Daten als *EAP-Response* zurück an den Authentifikator gesandt und von da aus in einem *RADIUS-Access-Request* weiter an den RADIUS-Server.

8. Falls dieser den Zugriff bewilligt, schickt er ein *RADIUS-Access-Accept*, das den Authentifikator ein *EAP-Success* an den WLAN-Client senden lässt. Der „logische Port" ist nun im *authorized* Zustand.

9. Der Access Point kann anschließend mittels *EAPoL-Key* einen dynamisch generierten Schlüssel an den WLAN-Client übermitteln. Dieser hat nun Zugriff auf das Netz und kann den übertragenen Schlüssel zur Verschlüsselung nutzen.

5.4.2 Schwäche von 802.1x für WLAN-Nutzung

Einen wesentlichen Schwachpunkt stellt die Tatsache dar, dass man eine mittels 802.1x zu einem „logischen Port" aufgebaute authentifizierte Relation nicht auf einen anderen „logischen Port" übertragen kann. Ein solcher „Umzug" kommt bei kabelgebundenen Ports nicht vor und wurde deshalb nicht beachtet. Durch diese Schwäche muss bei einem Wechsel des Access Points durch den WLAN-Client eine komplette Authentifizierung erfolgen, wodurch sich die Zeit für das Roaming (Wechsel zwischen den Access Points) wesentlich erhöht. Dies macht sich speziell bei zeitkritischen Anwendungen wie Audio/Video-Streaming negativ bemerkbar.

Dieser Schwachpunkt wurde beim Entwurf des IEEE 802.11i Standards beachtet. In 802.11i wurde das Schlüsselmanagement komplett überarbeitet (siehe Abschnitt 4.3.2.3). Damit besitzt sowohl der Access

Point als auch der WLAN-Client den passenden PMK (Pairwise Master Key) und kann daraus alle weiteren Schlüssel generieren.

Verlässt nun der WLAN-Client durch Roaming den Access Point, so bleibt dieses PMK eine gewisse Zeit im Speicher des Access Points. Falls der WLAN-Client durch einen erneuten Wechsel zu diesem Access Point zurückkehrt, muss er nur noch die Gültigkeit seines PMK nachweisen, hierzu tauschen die beiden Geräte lediglich vier Pakete (4-Way-Handshake) aus.

Ein weiteres Verfahren ist das sogenannte „Pre-Authentication"-Verfahren. Diese Methode ist ebenfalls im 802.11i standardisiert worden. Hierbei scannt der WLAN-Client im Hintergrund nach weiteren Access Points zu denen er eventuell wechseln könnte und führt dort bereits eine Authentifizierung nach 802.1x durch, wodurch beim Wechsel zu diesem dann ebenfalls nur noch der 4-Way-Handshake durchgeführt werden muss.

Für diese Verfahren benötigt man aktuelle Hardware (sowohl WLAN-Netzkarte als auch Access Points), die den kompletten 802.11i Standard bereits unterstützen.

Cisco hatte bereits vor diesem Standard eine eigene proprietäre-Methode mittels LEAP implementiert, die ein „schnelles" Roaming zwischen den Access Points ermöglicht.

5.5 Probleme beim EAPoL und EAPoW

Neben dem oben genannten Schwachpunkt des 802.1x Protokolls gibt es noch weitere Probleme bei der Authentifizierung mittels EAP. Gerade bei WLAN, bedingt durch das sogenannte „Shared Medium" (Luft), muss darauf geachtet werden, dass man eine möglichst sichere Authentifizierungsmethode wählt. Ansonsten ist es leicht möglich *Man-in-the-Middle*-Attacken durchzuführen. Aber auch sogenannte *Wörterbuch*-Angriffe (engl. dictionary hack) sind bei einigen Methoden möglich. Tabelle 5.4 zeigt einen Überblick zu möglichen Angriffsarten aller, für das WLAN relevanten, EAP-Verfahren.

Verfahren	Server Authentifizierung	Supplikant Authentifizierung	Angriffsmöglichkeiten	dynamische Schüssel
MD5	keine	Passwort-Hash	Man-in-the-Middle, Wörterbuch, Sessionübernahme	nein
TLS	Public Key Zertifikat	Public Key Zertifikat	keine bekannt	ja
TTLS	Public Key Zertifikat	PAP, CHAP, EAP-Verfahren	Man-in-the-Middle	ja
PEAP	Public Key Zertifikat	EAP-Verfahren	Man-in-the-Middle	ja
LEAP	Passwort-Hash	Passwort-Hash	Wörterbuch	ja

Tabelle 5.4: Überblick: EAP-Authentifizierungsverfahren

Auch trotz dieser Vorkehrungen können Angreifer versuchen den Betrieb des WLAN zu stören, indem sie DoS (Denial-of-Service)-Attacken in Form von gefälschten *EAP-Start-* bzw. *EAP-Logoff*-Paketen an Access Point und WLAN-Client schicken. EAPoL sieht keine Sicherung dieser Pakete durch einen Integritätscheck vor.

Auch könnte mit einer Flut gefälschter *EAP-Success*-Pakete versucht werden, den Verbindungsaufbau des WLAN-Clients zu verhindern bzw. den RADIUS-Server mit Authentifizierungsanfragen zu überlasten. Allerdings verhindert dies der Authentifikator durch eine Begrenzung der gleichzeitig möglichen Requests.

5.6 Betriebssystemunterstützung (Supplikant)

Normalerweise wird 802.1x im Treiber bzw. Betriebssystem selbst implementiert. Somit sind an der Hardware selbst keine Änderungen notwendig. Microsoft hat ab Windows 2000/XP 802.1x im Betriebssystem implementiert. Für ältere Betriebssysteme wie Windows 95/98/ME und Windows NT wurden durch Erweiterungen von verschiedenen Herstellern nachträglich eine 802.1x Unterstützung geschaffen.

Unter Linux gibt es zwei verschiedene Supplikanten: zum einen *xsupplicant* [35] und zum anderen *wpa_supplicant* [36]. Letzterer wird für WLAN-Support eingesetzt, da diese Software zusätzlich die WPA/WPA2 Unterstützung für verschiedene WLAN-Chipsätze bietet. Weiterhin gibt es von verschiedenen Herstellern kommerzielle Software mit Linux-Unterstützung zu diesem Thema.

Bei MacOS X wurde die 802.1x Unterstützung ebenfalls integriert, es gibt außerdem ebenfalls kommerzielle Lösungen.

Abschliessend soll Tabelle 5.5 eine Übersicht von Supplikanten für die verschiedenen Betriebssysteme geben:

Name	Hersteller	Verfahren	Systeme	dyn. WEP-Keys / WPA,802.11i
MS 802.1x Supplicant	Microsoft Inc.	MD5(LAN), TLS, PEAP	MS Windows XP	ja/ja
SecureW2	Alfa & Ariss	TTLS	MS Windows XP	ja/ja
Wire1x	Wireless Internet Research & Engineering (WIRE) Lab.	MD5, TLS, TTLS,PEAP	MS Windows XP/NT/2k/98/ME	nein/nein
xsupplicant	Open1x	MD5, TLS, TTLS, PEAP, LEAP	Linux	ja/nein
wpa_supplicant	Jouni Malinen	TLS, TTLS, PEAP, LEAP	Linux	ja/ja

Fortsetzung siehe nächste Seite

Mac 802.1x Supplicant	Apple	MD5, TLS, TTLS, PEAP, LEAP	MacOS X Panther 10.3.x	ja/ja
AEGIS	Meetinghouse Data Communications	MD5, TLS, TTLS, LEAP, PEAP	MS Windows XP/NT/2k/98/ME, Pocket PC, MacOS, Palm, Linux, Solaris	ja/ja
Odyssey	Funk Software Inc.	TLS, TTLS, PEAP, LEAP	MS Windows XP/2k/98/ME, Pocket PC, Windows Mobile	ja/ja

Tabelle 5.5: Überblick für Supplikant-Software

6 AAA - Authentication, Authorization, Accounting

Die Abkürzung AAA (gesprochen "Triple A") setzt sich aus Authentifikation (engl. Authentication), Autorisierung (engl. Authorization) und Abrechnung (engl. Accounting) zusammen.

Diese Begriffe werden im Folgenden als Synonym für folgende Definitionen verwendet:

- **Authentifikation** ist der Vorgang der Überprüfung einer angegebenen Identität. Das Kernstück bildet die Überprüfung der Identität anhand von Login und Passwort sowie Chip-Ausweise oder ähnlichen kryptographischen Techniken.
- **Autorisierung** ist der Vorgang zur Überprüfung von definierten personenbezogen oder gruppenbezogenen Nutzungsrechten für angebotene Dienste.
- **Abrechnung** ist die Sammlung von Daten auf der Grundlage der Ressourcennutzung, z. B. der übertragenen Datenmenge oder der Nutzungsdauer.

6.1 Allgemeine AAA-Architektur

Aufgrund des stetig wachsenden Bedarfs an derartigen Sicherheits- und Verwaltungsmechanismen entstand innerhalb der IETF 1995 eine Arbeitsgruppe, die *AAAarch Research Group*, die sich zum Ziel gesetzt hat, eine generelle AAA-Architektur zu schaffen, die allgemein und flexibel genug ist, so dass sie für beliebige Anwendungen verwendet werden kann.

Die AAA-Infrastruktur besteht aus AAA-Client und AAA-Servern, die untereinander in Verbindung stehen und ein spezielles AAA-Protokoll zur Kommunikation benutzen.

Der Client stellt hierbei eine Anfrage an den AAA-Server, der diese analysieren und die entsprechende Authentifizierungs-/Autorisierungsentscheidung treffen muss. Dazu besitzt er Regeln (logische und algebraische Formeln) sowie verschiedene Hilfskomponenten, die die notwendigen Informationen liefern können.

Abbildung 6.1: AAA-System-Architektur

Als Hilfskomponenten sind das ASM (Application Specific Module), das *Event Log* (Ereignisspeicher) und das *Policy Repository* (Regelwerk) zu nennen.

Das ASM stellt die Schnittstelle zu den anwendungsspezifischen Informationen dar. Das *Event Log* ist ein Datensystem, das alle auftretenden Ereignisse chronologisch speichert, die als Grundlage für spätere Entscheidungsfindungen dienen könnten. Das *Policy Repository* besitzt alle Auskünfte für die Autorisation von verfügbaren Diensten.

Falls ein AAA-Server nicht über die notwendigen Informationen verfügt, die für die Abarbeitung der Client Anfrage notwendig sind, muss er in der Lage sein, diese Anfrage an einen anderen AAA-Server weiterzuleiten. Diese wichtige Eigenschaft, dass ein AAA-Server gleichzeitig als Client und Server auftreten kann, nennt man Proxy-Fähigkeit.

Da es bereits verschiedene AAA-Implementierungen gibt, benötigt man eine Lösung, die die PDUs (Protocol Data Unit) des einen Protokolls in die PDUs des anderen umschreiben kann. Dies übernimmt ein sogenanntes AAA-Gateway.

6.2 AAA-Protokolle

6.2.1 Anforderungen

6.2.1.1 Authentifikation

Authentifizierung bedeutet, sicherzustellen, dass man tatsächlich mit der Instanz kommuniziert, die sie vorgibt zu sein. Dies bezeichnet man auch als „authentication of the channel endpoint".

Authentifizierung kann auf verschiedene Art und Weisen stattfinden. Die gebräuchlichste Art ist die Kombination aus Benutzername und Passwort. Allerdings ist sie unsicher, da der Anwender meist selbst das Passwort setzen kann und dies meist zu kurz und damit leicht entschlüsselbar ist. Andere Verfahren setzen dabei auf *Public Key* Kryptographie oder Chipkartensysteme.

Da die Authentifizierung viele verschiedene Sicherheitsanforderungen abdeckt, ist sie ein wichtiger Bestandteil bei sicherheitskritischen Diensten.

6.2.1.2 Autorisierung

Normalerweise folgt eine Autorisierungsprozedur einer bereits erfolgreich durchgeführten Authentifizierung. Sie ist gleichbedeutend mit der Frage, ob ein Nutzer ausreichende Rechte besitzt, um die von ihm geforderte Ressource zugeteilt zu bekommen oder nicht.

Die wesentlichen Anforderungen an die Autorisierungsprozeduren in AAA-Systemen lassen sich wie folgt zusammen fassen:

- Ein AAA-Protokoll sollte separate und muss kombinierte Authentifizierungs- und Autorisierungs-Nachrichten unterstützen.
- Ein AAA-Protokoll muss proxyfähig sein, d. h. es muss eine Benutzeranfrage an einen anderen AAA-Server zur Bearbeitung weiterleiten können.

Es existieren drei verschiedene Autorisierungssequenzen: *Agent*, *Pull* und *Push*.

Bei der *Agent*-Sequenz stellt der Benutzer seine Anfrage an den AAA-Server und dieser seinerseits an das „Service-Equipment", das den Dienst bereitstellt.

Bei der *Pull*-Sequenz stellt der Benutzer seine Anfrage direkt an das „Service-Equipment", das seinerseits den AAA-Server eine Anfrage zur Entscheidung stellt.

Bei der *Push*-Sequenz stellt der Benutzer seine Anfrage an den Server, der ihm bei bestandener Prüfung ein Ticket ausstellt, mit dem der Benutzer seine Anfrage an das „Service-Equipment" nach dem Dienst stellen kann.

6.2.1.3 Abrechnung

Bei der Abrechnung sammelt das System Daten über den Verbrauch von Ressourcen zu Analyse-, Planungs- und Abrechnungszwecken. In allen Anwendungsfällen, in denen dies eingesetzt wird, gibt es Sicherheitsanforderungen, die erfüllt sein müssen, damit dieser Dienst zuverlässig funktioniert.

Besonders wichtig ist die Unterscheidung, ob Abrechnung innerhalb einer administrativen Zone stattfindet (*Intra Domain Accounting*), oder ob die Zonengrenzen verlassen werden (*Inter Domain Accounting*). Da sich beim *Inter Domain Accounting* ein Teil der Datensammlung und -übermittlung der eigenen administrativen Kontrolle entzieht, ist diese Art wesentlich anfälliger gegenüber Verletzungen der Sicherheit.

Da die AAAarch Gruppe keinerlei Aussagen über das zu verwendende Transportprotokoll macht, muss insbesondere auch über die Wahrscheinlichkeit und Konsequenz von Paketverlusten nachgedacht werden.

Bei Abrechnung zur Trendanalyse oder Kapazitätsplanung sind Verluste bei der Datenaufzeichnung leicht in Kauf zu nehmen. Falls dies hingegen zur Gebührenerfassung (engl. Billing) genutzt wird, können schon geringe Verlustraten oder ein Totalausfall zu größeren finanziellen Schäden führen. In diesem Fall ist ein archivarischer Ansatz erforderlich, der es erlaubt, Paketverluste durch Wiederholungen zu kompensieren.

6.2.2 RADIUS

6.2.2.1 Geschichte

RADIUS (Remote Authentication Dial In User Service) ist ein AAA-Protokoll. Es wurde ursprünglich von der Firma Livingston Enterprises für ihren *Network Access Server* entwickelt (etwa 1989). Von Steve Willens wurde es 1992 erstmals spezifiziert. Um die Entwicklung eines offenen Standards zu unterstützen, gründete die IETF 1995 eine RADIUS-Arbeitsgruppe. Im April 1997 gab die IETF die Spezifizierung als Standard RFC 2138 [37] heraus. Ebenfalls im April 1997 wurde mit RFC 2139 [38] das RADIUS-Accounting spezifiziert. Die RFCs 2865 [39] und 2866 [39] lösten im Jahr 2000 die ursprünglichen Spezifizierungen ab.

Bis heute hat sich RADIUS als offene Lösung für alle Arten von Einwahl- (engl. Dial-In) bis zu Terminalzugängen etabliert.

6.2.2.2 Überblick

RADIUS beschreibt eine AAA-Architektur, die die AAA-Dienste verwaltungszonenübergreifend ermöglicht. Dabei entspricht das RADIUS-Protokoll dem Client-Server-Modell.

Der Client ist im Allgemeinen ein sogenannter NAS (Network Access Server), z. B. ein Switch oder Access Point. Der Client sendet an den Server die Authentifizierungsanforderungen. Entweder beantwortet der Server sofort selbst diese Anfrage oder er leitet sie in der Rolle des Proxys an einen anderen Server weiter. Dabei fungiert dieser gegenüber dem weiteren Server als Client. Der weitergeleiteten Nachricht kann der Proxy Attribute anhängen, die z. B. ermöglichen, die Anfrage als Weiterleitung zu identifizieren.

Der genaue Aufbau einer RADIUS-Nachricht wird im nachfolgenden Abschnitt beschrieben.

Die Nachrichten zwischen Client und Server werden per UDP übertragen. Dabei verzichtet man bewusst auf die Transportsicherheit von TCP. Der Grund dafür ist, dass die TCP-Mechanismen, die einem sicheren Transport dienen, nicht den Anforderungen von RADIUS genügten. Ist z. B. ein RADIUS-Server nicht erreichbar, so soll der Client die Anfrage nach einigen wiederholt fehlgeschlagenen Versuchen sofort einem alternativen Server zustellen. Um jedoch nicht von den Verzögerungen und Zeitbegrenzungen von TCP abhängig zu sein und diese selbst in weiten Grenzen manipulieren zu können, entschied man sich für UDP.

Im ursprünglich definierten RADIUS-Standard wurden die UDP Portnummern 1645 und 1646 für den RADIUS-Server definiert. Später stellte sich jedoch heraus, dass diese Portnummern bereits durch einen anderen Dienst in Benutzung waren. So wurden nachträglich die UDP Portnummern 1812 und 1813 für den RADIUS-Server verwendet.

Um eine gewisse Absicherung zwischen Client und Server aufzubauen, wurde ein entsprechendes gemeinsames Geheimwort (engl. Shared Secret) festgelegt. Dieses dient zur Authentifizierung des Client gegenüber den Server.

6.2.2.3 RADIUS-Paket-Form

Wie bereits beschrieben werden die RADIUS-Nachrichten per UDP versendet. Dabei ist genau eine Nachricht in einem UDP-Paket. Der allgemeine Aufbau einer solchen Nachricht ist in Abbildung 6.2 zu sehen und in Tabelle 6.1 näher beschrieben.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|     Code      |  Identifier   |            Length             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                                                               |
|                         Authenticator                         |
|                                                               |
|                                                               |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  Attributes ...
+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Abbildung 6.2: RADIUS-Paket-Form

Typ	Bytes	Beschreibung
Code	1	Bestimmt Typ des RADIUS-Pakets
Identifier	1	Identifiziert eine Anfrage
Length	2	Länge des Pakets in Byte, gebildet über gesamte Nachricht
Authenticator	16	Zufallszahl (im Request) bzw. MD5-Hash (im Response)
Attributes	x	Beliebige Anzahl an Tripel aus Typ, Länge und Wert

Tabelle 6.1: Aufbau eines RADIUS-Pakets

Code: Im Anhang in Tabelle 9.2 sind alle bisher definierten RADIUS-Codes aufgelistet.

Identifier: Der Identifikator ist im Request und Response gleich und ermöglicht so die Zuordnung von zusammengehörigen Request/Response-Paaren. Mehrfach gesendete Pakete können so vom RADIUS-Server erkannt werden.

Length: Die Länge des Pakets (in Byte) wird gebildet über: Code, Identifier, Length, Authenticator und Attribute. Daten im Paket außerhalb der in diesem Feld angegebenen Länge müssen ignoriert werden. Pakete, die kleiner sind als die Länge vorgibt, müssen verworfen werden.

Authenticator: In jedem Request wird vom Client eine möglichst unvorhersehbare 16 Byte lange Zufallszahl erstellt und in das Authenticator-Feld eingetragen. In jedem Response wird eine, durch Anwendung von MD5 über die zu schützenden Inhalte des Pakets gebildete, 16 Bytes große Zahl zurückgeliefert.

Attributes: Attribute sind Tripel, die aus Typ (1 Byte), Länge (1 Byte) und Wert (max. 256 Byte) bestehen (siehe Abbildung 6.3). Sie können spezifische Authentifikations- und Autorisationsinformationen sowie Konfigurationsdetails enthalten. Das Ende der Attributliste definiert das „Length"-Feld des RADIUS-Pakets. Es können beliebig viele Attribute in einer RADIUS-Nachricht vorkommen, auch wenn sie gleichen Typs sind. Die Attribute werden im RADIUS als AVP (Attribute Value Pairs) bezeichnet.

```
 0                   1                   2
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
|     Type      |    Length     |  Value ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Abbildung 6.3: RADIUS-Attribute

Die Attributtypen sind dabei in verschiedene Kategorien unterteilt. Eine Tabelle mit einem Überblick der verschiedenen Typen ist im Anhang (Tabelle 9.3) zu finden. Ebenfalls im Anhang befindet sich eine Auflistung verschiedener definierter RADIUS-Attribute (Tabelle 9.4). Ausführliche Informationen zu den Attributen sind in den RFCs 2865 bis 2869 zu finden.

6.2.2.4 RADIUS-Ablauf

Abbildung 6.4 zeigt einen typischen Ablauf des RADIUS-Protokolls zwischen Client (NAS) und Server:

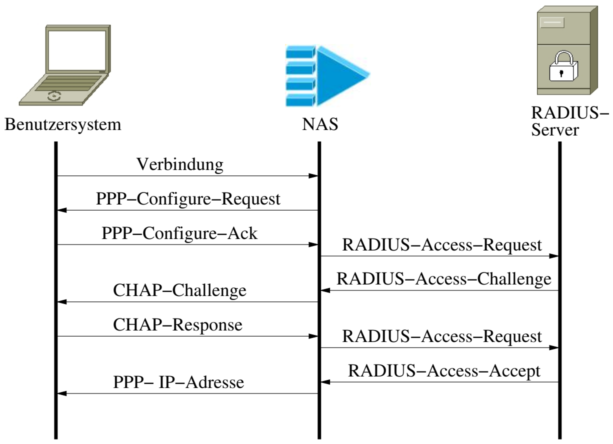

Abbildung 6.4: RADIUS-Attribute

1. Client sendet Authentifikationsanfrage
2. Server sendet eine Challenge (optional)
3. Client leitet Challenge an Benutzer weiter, antwortet mit Antwort des Benutzers
4. eventuell weitere Challenges
5. Server akzeptiert Anfrage oder lehnt sie ab

6.2.2.5 RADIUS-Authentifizierung

In der RADIUS-Spezifizierung wurde für die Authentifizierung PAP und CHAP bestimmt.

Bei PAP (RFC 1334 [24]) übergibt der RADIUS-Client das vom Supplikant ermittelte Benutzerkennzeichen und Passwort im Klartext an den Server. Hierfür werden die Attribute *User-Name* und *User-Password* genutzt. Dieses Verfahren gilt als unsicher, weil das Passwort im Klartext übermittelt wird.

Beim CHAP (RFC-1994 [25]) wird mehr Wert auf die Sicherheit der Passwörter gelegt. Hierzu generiert der RADIUS-Client eine Zufallszahl (16 Bytes) und sendet sie an den Nutzer. Dieser antwortet darauf mit einer CHAP-ID, dem Benutzerkennzeichen und dem aus der Zufallszahl und dem Passwort mittels MD5 generierten CHAP-Passwort. Der RADIUS-Client schickt dies inklusive der generierten Zufallszahl an den RADIUS-Server. Der RADIUS-Server ermittelt nun anhand der in seiner Datenbasis vorhandenen Login/Passwort-Kombination, der übermittelten Zufallszahl und CHAP-ID ebenfalls das CHAP-Passwort. Es folgt ein Vergleich des übersendeten und generierten CHAP-Passworts. Daraufhin reagiert der Server entweder mit einer *Access-Accept*- oder *Access-Reject*-Nachricht.

Da bei PAP das Klartext-Passwort an den Server übermittelt wird, können die Passwörter in dessen Datenbasis auch verschlüsselt hinterlegt sein. Mittels Anwendung des entsprechenden Krypto-Verfahrens kann der Server dann das übermittelte Passwort mit dem verschlüsselt hinterlegten Passwort vergleichen. Dies geht bei CHAP nicht. Bei sogenannten *Challenge-Response*-Verfahren, wie es CHAP ist, muss auf beiden Seiten das Passwort im Klartext hinterlegt sein.

6.2.2.6 RADIUS und EAP

Mit RFC 3579 [40] wurde im Jahr 2003 die RFC 2869 [41] um eine zusätzliche Authentifizierungsmethode erweitert. Diese Spezifikation beschreibt, wie EAP-Pakete vom RADIUS-Client an den RADIUS-Server verpackt und versendet werden. RFC 3580 [42] beschreibt zusätzlich den Umgang von RADIUS mit 802.1x.

Hierfür wurden zwei neue Attributtypen eingeführt: *EAP-Message* und *Message-Authenticator*.

Die EAP-Nachrichten vom Supplikant werden im neuen *EAP-Message*-Attribut verschickt. Falls die EAP-Nachrichten länger als die maximale RADIUS-Attributlänge (256 Byte) sind, wird diese EAP-Nachricht in mehrere AVPs aufgeteilt und versendet. Längere EAP-Dialoge werden in *Access-Request*- und *Access-Challenge*-Nachrichten transportiert.

Der Client kann die EAP-Pakete „unbesehen” an den Server durchreichen. Er muss allerdings die *EAP-success/failure* erkennen und entsprechend reagieren. Der Server muss die entsprechenden EAP-Methoden unterstützen.

Jede Nachricht, die ein EAP-Attribut enthält, muss signiert werden. Diese Signatur hat den Attributtyp *Message Authenticator* und ist eine HMAC (Hash Message Authentication Code)-MD5 Prüfsumme (RFC 2104 [43]) über die gesamte Nachricht. Der gemeinsame Schlüssel von Client und Server wird als Schlüssel für das HMAC-Verfahren verwendet.

Die Abbildungen 5.6 und 5.7 aus Kapitel 5 zeigten bereits den vollständigen Ablauf einer Authentifizierung mittel EAP und RADIUS.

6.2.2.7 RADIUS-Accounting

Unter dem RADIUS-Accounting versteht man die Abrechnung der durch den Client erbrachten Services. Hierbei werden die zwei RADIUS-Codes *Accounting-Request* und *Accounting-Response* benutzt. Es werden 11 mögliche Attribute, z. B. *Account-Status-Type*, zur Verfügung gestellt.

Accounting-Start: Client sendet an den Server einen *Accounting-Request*, der unter anderem folgendes beinhaltet:

- Welcher Service erbracht wird
- An welchen Benutzer der Service erbracht wird

Accounting-Stop: Client sendet an den Server einen *Accounting-Request*, der u. a. beinhaltet:

- Welcher Service erbracht wurde
- An welchen Benutzer der Service erbracht wurde
- Optionale Angaben: Zeit, Datenvolumen

Die genauen Informationen zum RADIUS-Accounting sind in RFC 2866 [44] zu finden.

Die RADIUS-Erweiterungen aus RFC 2869 [41] beschreiben u. a. die Möglichkeit einer „Zwischenabrechnung" (engl. Interim Accounting Updates). Dabei kann der Client in frei definierbaren Zeitabständen die aktuellen Zwischenwerte an den Server schicken.

6.2.2.8 RADIUS-Tunnel

Die RFC 2868 [45] und die oben bereits erwähnte RFC 2869 [41] beschreiben RADIUS-Erweiterungen. Dabei wird auch ein *Tunnel*, der Datenverbindungen durch den RADIUS-Server leitet, beschrieben. Dies findet aber sehr selten Anwendung und soll hier nur der Vollständigkeit halber erwähnt werden.

6.2.2.9 Schwächen des RADIUS-Protokolls

Nachdem die RADIUS-Implementierungen seit über zehn Jahren im Einsatz sind, zeigen sich die Schwächen des Protokolls, die eine Nutzung unter gestiegenen Anforderungen in den Punkten Transportsicherheit, Skalierbarkeit und Flexibilität in Frage stellen. Nachfolgend sollen kurz einige Problemstellungen erläutert werden:

- Das Transportprotokoll UDP stellte sich als Schwachstelle heraus. Das RADIUS-Protokoll beschreibt eine Strategie, wie beim Ausbleiben einer Antwort vom Server zu reagieren ist, nämlich mit Wiederholungsversuchen nach einem Timeout. Doch weder die Anzahl der Wiederholungen, noch die Zeitintervalle sind im Protokoll festgelegt. So kann es zwischen den verschiedenen Implementierungen zu Inkompatibilitäten kommen.
- Der Client kann lediglich an einem Timeout erkennen, dass der Server nicht erreichbar ist. Der Server sendet an den Client keine Status- oder Fehlermeldung. Somit ist weder auf der Transportschicht noch auf höherer Ebene eine Fehlererkennung implementiert. So können Fehler wie z. B. lange Antwortzeiten vom Server, weil dieser auf eine langsame Datenbasis zugreift, vom Client als Fehler (Server-Timeout) gewertet werden. Er wiederholt seine Anfrage. Tritt ein Übertragungsfehler zwischen Proxy und Server auf, so reagiert nicht der Proxy mit einer Wiederholung, sondern der Client stellt die Anfrage erneut, wodurch ein unnötig hoher Datenverkehr in Kauf genommen werden muss.
- Es ist keine Fehlermeldung bei fehlerhaften Nachrichten (Länge oder Typ der Nachricht falsch) vom Server an den Client vorgesehen. Der Client versucht eventuell vergeblich diese falsche Nachricht weiterhin zu senden. Der Server hingegen ignoriert diese Nachrichten einfach.
- RADIUS ist nicht modular erweiterbar. Es lassen sich zwar weitere Attribute definieren, doch insgesamt sind nur 256 verschiedene erlaubt bzw. möglich. Implementieren verschiedene Hersteller neue Funktionen und Attribute, so kann es zu Überschneidungen in dem begrenzten Namensraum und daher zu Inkompatibilitäten kommen. Weiterhin ist die Länge des AVP durch das lediglich 1 Byte große Längenfeld auf 256 Bytes begrenzt. Sollten größere Datenmengen versendet werden, so müssen sie auf mehrere AVPs verteilt werden.
- RADIUS ist als Client/Server-Modell implementiert, somit ist keine Möglichkeit vorgesehen, eine Kommunikation mit dem Client durch den Server zu beginnen. Gerade bei der Abrechnung ist es aber eigentlich notwendig bzw. üblich, dass der Server diese Informationen abfragt oder den Client auffordert, die Sitzung (engl. session) eines Nutzers zu beenden.

6.2.2.10 Implementierungen

Da der RADIUS schon seit vielen Jahren auf dem Markt zu finden ist, gibt es eine Vielzahl an Implementierungen. So gibt es neben den RADIUS-Server von kommerziellen Firmen auch verschiedene Open-Source-Lösungen.

Nachfolgend einige RADIUS-Implementierungen:

- Cisco Secure Access Control Server [46]
- FreeRADIUS [47]

6.2.3 DIAMETER

DIAMETER steht nicht wie RADIUS als Abkürzung, sondern soll eigentlich nur verdeutlichen, dass es sich hierbei um RADIUS Version 2 handelt. DIAMETER wurde nicht von Grund auf neu entwickelt, ist aber auch nicht als Erweiterung von RADIUS zu sehen, sondern eher als verbesserte „Neuversion”.

6.2.3.1 Geschichte

Um den im Abschnitt 6.2.2.9 beschriebenen Problemen und Schwächen des RADIUS-Protokolls zu begegnen, wurde bereits 1996 von Pat Calhoun mit der Entwicklung von DIAMETER begonnen.

Im weiteren Verlauf beschäftigten sich verschiedene Arbeitsgruppen mit den Spezifizierungen der Anforderungen an das Design von DIAMETER.

Bei der Entwicklung wurde besonders auf Kompatibilität mit RADIUS Wert gelegt, um schrittweise eine Migration zu DIAMTER zu erleichtern.

Ende 2003 wurde durch die IETF das *DIAMETER Base Protocol* durch die RFC 3588 [48] veröffentlicht. Im August diesen Jahres wurden durch die RFCs 4004 [49], 4005 [50], 4006 [51] und 4072 [52] die sogenannten *Applications Base Protocols* spezifiziert, die auf das Basis-Protokoll aufsetzen. Weitere Erweiterungen sind zu erwarten.

6.2.3.2 Überblick

DIAMETER setzt sich aus einem allgemeinem Teil, dem Basis-Protokoll, und aus verschiedenen anwendungsspezifischen Erweiterungen, den sogenannten Applikation-Protokollen, zusammen.

Das Basis-Protokoll beschreibt den Transport der Daten, Fehlermeldungen und grundlegende Sitzungs- und Abrechnungsfunktionalitäten.

Die Applikation-Protokolle stellen dann die Unterstützung für bestimmte Technologien, wie z. B. Mobile IP und Dial-In-Zugänge, zur Verfügung.

Durch das Konzept der Applikationen ist DIAMETER im Gegensatz zu RADIUS modular erweiterbar. Bisher sind vier Applikation-Protokolle spezifiziert, die *Mobile IPv4 Application*, die *Network Access Server Application*, die *Credit-Control Application* und die *CMS (Cryptographic Message Syntax) Application*, die eng mit dem Basis-Protokoll verbunden ist und die Sicherheit aller Applikationen gewährleistet.

Trotz der unterschiedlichen Funktionalitäten der verschiedenen Applikationen müssen sie alle das DIAMETER-Basis-Protokoll unterstützen.

Abbildung 6.5 zeigt symbolisch den grundlegenden Aufbau:

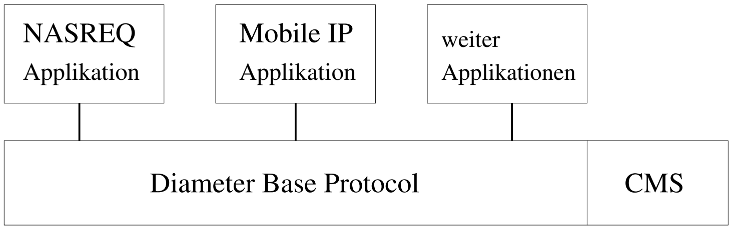

Abbildung 6.5: DIAMETER-Protokoll

6.2.3.3 Komponenten und Architektur

Jede DIAMETER-Instanz wird zunächst als Knoten bezeichnet. Dabei gibt es verschiedene Komponenten:

Client Als *Client* werden die Geräte an der „Grenze” des Netzwerkes bezeichnet, die für die Zugangskontrolle zuständig sind, z. B. Access Point oder Switch.

Server Der *Server* ist die Komponente, der die Authentifikations-, Autorisations- und Abrechnungsanfragen verarbeitet.

Relay Agent Der *Relay Agent* leitet Nachrichten anhand von Routing Informationen in den Nachrichten und der eigenen internen Routingtabelle weiter. Dabei kann der *Relay Agent* der Nachricht Routinginformationen hinzufügen oder aus ihr entfernen. Ansonsten arbeitet er transparent.

Proxy Agent Der *Proxy Agent* leitet genau wie der *Relay Agent* die Nachrichten weiter, kann dabei aber ungültige Anfragen mit entsprechenden Fehlermeldungen und *Access-Reject*-Meldungen abfangen. Der *Proxy Agent* kann also den Inhalt der Nachrichten auswerten und verarbeiten. Damit dies funktioniert, muss die Komponente das verwendete Applikations-Protokoll unterstützen.

Redirect Agent Der *Redirect Agent* ist für die Kommunikation zwischen Client und Server verantwortlich. Auf Anfrage des Clients sendet der *Redirect Agent* die Zieladresse des gewünschten Zielservers, so dass Client und Server eine direkte Kommunikation ermöglicht wird.

Translation Agent Der *Translation Agent* stellt ein Gateway dar, das AAA-Protokollnachrichten transferieren kann. Hiermit ist es z. B. möglich zwischen RADIUS und DIAMETER zu „übersetzen”. So soll die Migration von einem zum anderen Protokoll vereinfacht werden. Damit können bereits die DIAMETER-Server eingesetzt und die große Anzahl an Clients dann Schritt für Schritt umgerüstet werden.

Im Gegensatz zum RADIUS-Protokoll wurde das Client-Server-Modell zugunsten eines Peer-to-Peer-Modells im DIAMETER aufgegeben. Somit reagiert der Server nicht nur auf Anfragen der Clients, sondern kann ihnen gegebenenfalls Nachrichten senden. Damit werden einige Erleichterungen geschaffen, so z. B. ein Sessionmanagement, in dem der Server eine Verbindung beendet oder Abrechnungsinformationen gezielt vom Client abfragt.

Ein Beispiel für eine DIAMETER-Kommunikation wäre ein Access Point, der den Client darstellt und seine generierten AAA-Nachrichten über einen *Proxy Agent* an einen Server sendet.

Dabei werden die Nachrichten nicht mehr, wie beim RADIUS, mittels UDP übermittelt, sondern es wird auf TCP bzw. SCTP (Stream Control Transmission Protokoll; RFC-2960 [53]) aufgesetzt. Dies trägt zur Fehlervermeidung auf der Transportebene bei, wodurch ein sicherer Transport ohne Datenverlust oder doppelt ankommende Pakete gewährleistet wird.

Aber nicht nur auf der Transportschicht ist eine Fehlererkennung möglich, sondern auch auf der Anwendungsebene. So muss, bis auf wenige Ausnahmen, jeder Request beantwortet werden. Dazu enthalten alle Responses einen *Result-Code-AVP*, in dem ein Fehler- oder Erfolgsstatus übergeben wird.

Durch diese Fehlermeldungen im Protokoll ist es dem Client und dem *Proxy Agent* möglich, angemessen darauf zu reagieren. Im RADIUS-Protokoll obliegt die Reaktion auf Fehler allein dem Client. Im DIAMETER kann bereits der *Proxy Agent* darauf reagieren und so z. B. im Falle eines Serverausfalls vor Ablauf des Client-Timeouts einen Wiederholungsversuch starten oder die Nachricht direkt an einen alternativen Server schicken.

6.2.3.4 DIAMETER-Nachrichten-Aufbau

Der Aufbau der Nachrichten ist dem von RADIUS ziemlich ähnlich, so werden auch hier alle Nachrichten als AVP versendet. Allerdings wurde die Beschränkung von 256 verschiedenen AVPs im DIAMETER auf 2^{32} erhöht. Diese dürfen jetzt bis zu 2^{24} Bytes groß sein.

Abbildung 6.6 zeigt den allgemeinen Aufbau eines DIAMETER-Pakets:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |                 Message Length                |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| command flags |                  Command-Code                 |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                         Application-ID                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Hop-by-Hop Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      End-to-End Identifier                    |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|  AVPs ...
+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```

Abbildung 6.6: DIAMETER-Paket-Form

DIAMETER-Nachrichten bestehen aus einem statischen Header, der allgemeine Informationen über die Nachricht beinhaltet und den AVPs, die die eigentlichen Nutzdaten enthalten.

Version: Dieses 8 Bit Feld zeigt die DIAMETER-Version an. Aktuell gibt es nur Version 1.

Message Length: Hier wird die Länge (in Byte) der Nachricht inklusive Header in 24 Bit festgehalten.

Command Flags: Dieses Feld besteht aus 8 Bit. Durch Setzen bestimmter Bits wird angezeigt, dass es sich z. B. um einen Request oder Response handelt.

Command-Code: Dieses 24 Bit lange Feld beschreibt die Methode der Nachricht.

Application-Id: Dieses 32-Bit-Feld gibt an, zu welcher DIAMETER-Applikation diese Nachricht gehört.

Hop-By-Hop Identifier: Dieses 32-Bit-Feld wird von den Proxys zur Zuordnung der Nachrichten verwendet.

End-To-End Identifier: Dieses 32-Bit-Feld dient dem Sender und Empfänger der Nachricht zur korrekten Zuordnung.

6.2.3.5 DIAMETER-AVP-Aufbau

Dem Header einer DIAMETER-Nachricht folgen ein bzw. mehrere AVPs. In ihnen werden die eigentlichen Informationen übertragen. Dabei sind je nach Nachrichtentyp unterschiedliche AVPs vorhanden. Sie verfügen über einen identischen Headeraufbau.

Abbildung 6.7 zeigt den allgemeinen Aufbau eines AVP-Header:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           AVP Code                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|V M P r r r r r|                  AVP Length                   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                        Vendor-ID (opt)                        |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Data ...
+-+-+-+-+-+-+-+-+
```

Abbildung 6.7: DIAMETER-AVP-Header-Format

Code: Der 32 Bit lange Code zusammen mit der optionalen *Vendor-Id* identifiziert das AVP eindeutig. Der Bereich 0 bis 255 ist für die Abwärtskompatibilität zu RADIUS reserviert, der Bereich darüber spezifiziert die DIAMETER-AVPs.

Flags: Dieses Feld besteht aus 8 Bit und enthält Informationen über das AVP, z. B. ob es sich um ein herstellerspezifisches AVP handelt, ob es notwendig oder optional ist, die Notwendigkeit zur Verschlüsselung usw.

Length: Das 24 Bit große Feld beschreibt die Länge (in Byte) des AVPs inklusive Header.

Data: AVPs können unterschiedliche Typen von Daten transportieren. Datentyp kann u. a. OctetString, Integer64, Float32 sein. Weiterhin gibt es noch abgeleitete Datentypen, die aus den Basistypen abgeleitet sind, z. B. UTF8String und DIAMETERIdentity. Eine ausführliche Auflistung befindet sich im Anhang (Tabellen 9.5 und 9.6).

Einige der wichtigsten AVPs, die unter anderem im Basis-Protokoll spezifiziert sind:

- Session-Id
- Origin-Host
- Accounting-Record-Type
- Accounting-Record-Number
- Authorization-Lifetime
- Auth-Grace-Period
- Auth-Session-State
- Accounting-Realtime-Required
- Result-Code
  - 1000 - 1999 : Informeller Result-Code
  - 2000 - 2999 : Erfolgreicher Result-Code
  - 3000 - 3999 : Protokoll-Fehler
  - 4000 - 4999 : Vorübergehender Fehler
  - 5000 - 5999 : Permanente Fehler

6.2.3.6 DIAMETER-Zustandsautomat

Anders als im RADIUS-Protokoll sind im DIAMETER-Protokoll endliche Zustandsautomaten für die Verbindung, Authentisierung und Autorisierung sowie für die Accountingfunktionalität spezifiziert.

Der Automat für das Verbindungsmanagement beschreibt im Wesentlichen den Austausch beim Verbindungsaufbau/-abbau. Für die Authentisierung und Autorisierung existiert im DIAMETER-Protokoll für Client und Server jeweils ein zustandsbehafteter und ein zustandsloser Automat, der die Abläufe genau spezifiziert. Dabei bedeutet zustandsbehaftet, dass der jeweilige Knoten sich den Zustand merkt, wohingegen der zustandslose keine Zustände speichert und somit nach der Bearbeitung die Anfrage vergisst.

- **Zustandsbehafteter Client:** Ein zustandsbehafteter Client legt für die verschiedenen authentifizierten Nutzer und deren autorisierten Dienst Sitzungsdaten an. Diese Sitzungsdaten können durch verschiedene Faktoren verändert werden, so z. B. durch Ablauf eines Session-Timers oder durch Abmelden des Nutzers. Unmittelbar nach Beenden der Sitzung schickt der Client eine entsprechende Benachrichtigung an den Server.

- **Zustandsbehafteter Server:** Nach erfolgreicher Authentifizierung wechselt der Server für diesen Nutzer in einen neuen Zustand, indem er Sitzungsdaten für die Applikationen und Nutzer anlegt. Diese Sitzung kann dann durch verschiedene Ereignisse beeinflusst oder beendet werden. Falls der bereitgestellte Dienst abgebrochen werden soll, sendet der Server an den Client eine entsprechende Nachricht, die den Client zum Abbruch der Sitzung veranlasst.

- **Zustandsloser Client:** In diesem Fall sendet der Client seine Anfragen an den Server ohne dabei Sitzungsdaten zu speichern; er reagiert lediglich auf die Antworten des Servers.

- **Zustandsloser Server:** Ein zustandsloser Server überprüft die Authentifizierungsanfragen anhand der in der Anfrage befindlichen Daten und seiner Datenbasis. Er sendet eine entsprechende Antwort. Sitzungsdaten werden nicht gespeichert.

Für die Abrechnungsfunktionalität existiert auf der Seite des Clients nur ein zustandsbehafteter Automat, auf der Seite des Servers sowohl ein zustandsloser als auch ein zustandsbehafteter Automat. Der zustandsbehaftete Server hat dazu einen Timer, in dem er entsprechende Abrechnungsdaten erwartet und beim Ausbleiben dieser entsprechend reagieren kann. Der zustandslose Server speichert hingegen nur die vom Client empfangenen Werte. Beim Client werden für die Sitzung entsprechende Zustände und Daten gespeichert. So kann dieser auch auf entsprechende Fehlerzustände reagieren.

6.2.3.7 DIAMETER-Identity

Für jeden DIAMETER-Prozess, der auf einem Knoten generiert bzw. konfiguriert wurde, gibt es eine sogenannte *DIAMETER-Identity*. Diese ist im URI (Uniform Resource Identifier)-Syntax-Format gehalten und beinhaltet den FQDN (Fully Qualified Domain Name), den Port an dem der Dienst Verbindungen erwartet, das Transportprotokoll (TCP oder SCTP), das AAA-Protokoll (hier DIAMETER) und die Transportsicherheit (z. B. *none* oder *TLS*).

Hier eine beispielhafte *DIAMETER-Identity*:
aaa://host.abc.com:1812;transport=tcp;protocol=diameter

6.2.3.8 DIAMETER und EAP

Die Nutzung von 802.1x/EAP mit DIAMETER funktioniert ähnlich wie im RADIUS-Protokoll. Hierfür wurde eine DIAMETER-Applikation im RFC 4072 [52] spezifiziert. Diese basiert auf der Applikation NASREQ (Network Access Server Application) und hat deshalb eine ähnliche Funktionalität.

Die EAP-Kommunikation verläuft direkt vom EAP-Client zum Server und ist mittels der CMS-Applikation während des Datentransports gesichert.

Für die EAP-Kommunikation wurden entsprechende AVPs definiert. Somit lässt sich EAP in DIAMETER-Nachrichten kapseln und zur Authentifizierung und Autorisation nutzen.

Unter anderem gibt es folgende Attributtypen:

- DIAMETER-EAP-Request
- DIAMETER-EAP-Request
- EAP-Payload
- EAP-Reissued-Payload
- EAP-Master-Session-Key
- EAP-Key-Name
- Accounting-EAP-Auth-Method
- usw.

6.2.3.9 Vorteile von DIAMETER gegenüber RADIUS

Verbesserter Datentransport

- DIAMETER setzt auf den zuverlässigen Datentransport mittels TCP oder STCP.
- Verlorene Datenpakete können von jedem Knoten aus wiederholt übertragen werden.
- Für einen schnellen „Failover" wird auf persistente Verbindungen mit sogenannten „Heartbeat Messages" (auch „Watchdog Message" genannt) gesetzt.

Verbesserte Proxy-Funktionalität

- Durch eine *Hop-by-Hop* Fehlererkennung kann im Fehlerfall schnell reagiert und ein alternativer Weg zum Ziel gefunden werden.
- Ein Proxy kann automatisch eine wiederholte Übertragung veranlassen, falls ein Fehlerfall vorliegt.

Verbesserte Sitzungskontrolle

- Der Server kann mittels Benachrichtigung des DIAMETER-Clients eine Sitzung beenden.
- Der Server kann ebenfalls eine Re-Authentifizierung/-Autorisierung vom Nutzer veranlassen.

Verbesserte Datensicherheit

- Sicherung der *Hop-to-Hop* Verbindungen ist mittels IPSec oder TLS möglich und fester Bestandteil der Spezifikation.
- Die *End-to-End* Sicherheit schützt die Integrität und Vertraulichkeit der sensiblen AVPs über die verschiedenen Zwischenstationen hinweg.

6.2.3.10 Nachteile von DIAMETER gegenüber RADIUS

Die größten Nachteile von DIAMETER sind die fehlenden vollständigen Implementierungen sowie die noch ausstehende Unterstützung in den Netzkomponenten. Durch mangelnden praktischen Einsatz sind noch keine Nachteile ersichtlich.

Weiterhin ist DIAMETER wesentlich komplexer als das RADIUS-Protokoll, was einen größeren administrativen Aufwand nach sich ziehen kann.

6.2.3.11 Implementierungen

Bisher existieren nur sehr wenige Implementierungen, die auch nur einen kleinen Teil der Spezifizierungen unterstützen. Da erst Mitte 2005 die Spezifizierungen für die DIAMETER-Applikation-Protokolle erschienen sind, fehlen dazu meist die Implementierungen und/oder sind bisher nur teilweise realisiert.

Nachfolgend einige DIAMETER-Implementierungen:

- OpenDiameter [54]
- HP-UX Mobile AAA Server [55]
- DISC - DIameter Server Client (letzte Aktualisierung November 2003) [56]

6.2.4 TACACS - Terminal Access Controller Access Control System

6.2.4.1 Geschichte

Beim TACACS handelt es sich um ein Authentifizierungs- und Autorisations-Protokoll. Entwickelt wurde das Protokoll bereits vor 1990, aber erst im Juli 1993 durch die RFC 1492 [57] als Standard spezifiziert. Unter der Überschrift: *An Access Control Protocol, Sometimes Called TACACS* beschreibt sie TACACS als Abkürzung für *Terminal Access Controller Access Control System*.

Ursprünglich wurde der Standard zur Authentifizierung an Terminal-Servern entwickelt und später vereinzelt als Zugangs-Kontroll-System für TCP/IP-Netzwerke eingesetzt.

6.2.4.2 Architektur

Bei TACACS handelt es sich um ein Client/Server-Protokoll. Es basiert auf UDP (Port 49) und besitzt keinerlei Sicherheits- oder Verschlüsselungsverfahren. Es existieren nur genau zwei Nachrichtentypen: Request und Reply/Result.

Dabei hat das Request-Paket folgenden Aufbau:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |      Type     |             Nonce             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| User len/Resp | PW len/Reason |           data...             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Abbildung 6.8: TACACS-Request-Paket-Format

Das Reply/Result-Paket hat folgenden Aufbau:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |      Type     |             Nonce             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|   User len    | Password len  |           data...             |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Abbildung 6.9: TACACS-Response-Paket-Format

Folgende Requesttypen sind definiert:

- LOGIN
- SUPERUSER
- LOGOUT

Weitere Ausführungen zum Aufbau und zur Funktionalität sind u. a. auf den Webseiten von *The Internet NG Project* [58] zu finden.

6.2.4.3 Implementierungen

Das TACACS Protokoll wird heute nicht mehr so eingesetzt, allerdings noch aus Kompatibilitätsgründen unterstützt.

Folgende Serverimplementierungen sind verfügbar:

- Radtac Manager Server [59]
- HandyTacacs 1.0 [60]

6.2.4.4 XTACACS - eXtended Terminal Access Controller Access Control System

XTACACS steht für *eXtended Terminal Access Controller Access Control System* und ist eine Erweiterung des eigentlichen TACACS-Protokolls. Es wurde 1990 von Cisco eingeführt und besitzt, neben den Eigenschaften die TACACS bietet, zusätzlich noch die Möglichkeit der Abrechnung. Allerdings wurde dieses Protokoll nie durch eine RFC spezifiziert und ist damit auch nur als de facto-Standard anzusehen.

Ausführliche Informationen zum XTACACS-Protokoll sind ebenfalls bei *The Internet NG Project* [61] nachzulesen.

6.2.4.5 TACACS+

TACACS+ ist eine komplette Neuimplementierung des TACACS-Protokolls. Es wurde von Cisco entwickelt und ist nicht zu den TACACS- und XTACACS-Protokollen kompatibel. Es stellt eine komplette AAA-Umgebung mit Authentifizierung, Autorisierung und Accounting bereit.

Auch TACACS+ ist ein Client/Server Protokoll, das aber nicht wie seine Vorgänger auf UDP sondern auf TCP (TCP-Port 49) aufsetzt. Es unterstützt Verschlüsselung zwischen Client und Server (mittels PSK). Folgende Authentifizierungsprotokolle werden u. a. unterstützt: PAP, CHAP, MSCHAP.

Dieses Protokoll ist von Cisco lizenziert. Genauere Informationen zum Protokoll finden sich im IETF Draft draft-grant-tacacs-02 [62] oder bei *The Internet NG Project* [63].

Als wichtigste Serverimplementierung ist dabei der *Cisco Secure Access Control Server* zu nennen.

6.3 Weitere mögliche Ansätze

Es gibt verschiedene Ansätze, die sich nur auf die Authentifizierung der Nutzer als Zugangskontrolle beschränken. Die nachfolgenden Abschnitte beschreiben einige möglich Ansätze.

6.3.1 Webauthentifizierung

Mirko Parthey beschreibt in seiner Diplomarbeit "*Zugangsmanagement für Wireless LAN*" [64] vom Dezember 2001 den transparenten Zugang zum Campusnetz der TU Chemnitz und zum Internet für mobile Geräte. Hier beschreibt er die Authentifizierung der Nutzer mittels HTTP-Authentifizierung und dem HTTPS-Protokoll.

Zitat: „*Der Nutzer meldet sich für jede Sitzung mit Nutzerkennzeichen und Passwort an, dabei werden seine MAC- und IP-Adresse zeitweilig registriert. Während der Sitzung erfolgt nur eine schwache Authentifizierung auf Basis dieser Daten. Die Erreichbarkeit des Nutzerrechners wird regelmäßig geprüft und nach einer gewissen Zeit der Nichterreichbarkeit wird die Registrierung wieder gelöscht. Damit ist die unberechtigte Nutzung eines registrierten Zugangs deutlich erschwert.*"

Die hierfür erforderlichen Dienste werden mittels der von ihm programmierten Programme angepasst. Eine Firewall auf dem zentralen Zugangsserver, die ebenfalls mittels dieser Programme konfiguriert wird, regelt dabei die Zugangskontrolle.

Das Problem dieser Lösung ist die schlechte Skalierung. Durch die Auftrennung der WLAN-Bereiche in verschiedene Routingbereiche im Campusnetz der TU Chemnitz konnte diese Lösung nicht weiter verwendet werden und wurde ersetzt.

Diese Authentifizierungsmethode wurde bereits ausführlich untersucht und wird deshalb an dieser Stelle nicht weiter vertieft.

6.3.2 Authentifizierung mittels Applikationen

Neben der Authentifizierung wie im vorigen Abschnitt beschrieben, gibt es noch eine Reihe anderer denkbarer Möglichkeiten. Ein Ansatz wäre, auf die Authentifizierung bestimmter Anwendungen aufzusetzen. Erst nach erfolgreicher Authentifizierung an diesem Dienst wird die Netzanbindung für weitere Datenverbindungen aktiviert.

6.3.2.1 Authentifizierung mittels VoIP - SIP

Idee: Ein Beispiel hierfür wäre VoIP (Voice over IP) mittels SIP (Session Initiation Protokoll). Die Idee dahinter ist, die Nutzerdaten, die bereits für das VoIP existieren, für die Authentifizierung am Netzwerk zu nutzen. Dazu ist der Zugang zum Netzwerk soweit abzusichern, dass ausschließlich der für die Authentifizierung notwendige Datenverkehr stattfinden kann. Eine einheitliche Regelung ist nur in einer homogenen Netzinfrastruktur möglich und dabei auch nur durch die vom Hersteller gegebenen Mittel wie ACLs oder Paket-Filter.

Das 3GPP (3rd Generation Partnership Project) beschäftigt sich u. a. mit Themen wie SIP-Authentifizierung und Zugang zu drahtlosen und IP-Netzwerken. Ausführliche Informationen dazu finden sich in den durch den Verlag *Wiley* erschienenen Büchern *The 3G IP Multimedia Subsystem* [65] und *Convergence Technologie for 3G Networks* [66].

Probleme: Für die Nutzung einer anwendungsbasierten Authentifikation muss bereits eine TCP/IP-Verbindung aufgebaut sein. Eine Absicherung der Datenverbindung (besonders bei WLAN sinnvoll) muss getrennt von der Authentifizierung stattfinden und bedeutet zusätzlichen Aufwand. Die Zugangsbeschränkung vor der korrekten Authentifizierung muss nach Möglichkeit direkt auf dem Gerät stattfinden, an dem sich der Nutzer anmeldet, wodurch man auf herstellerspezifische Verfahren angewiesen ist.

Weiterhin unterliegt dieses Verfahren den Schwächen und Fehlern des Applikation-Protokolls sowie der möglichen Manipulation Dritter durch fehlende oder mangelnde Datensicherheit. Vor allem bei Mehrbenutzersystemen gibt es weitere denkbare Probleme.

6.3.2.2 Authentifizierung mittels IM (Instant Messaging)

Analog der im vorigen Abschnitt beschriebenen Methode der Authentifizierung mittels VoIP-Software, ist eine Authentifizierung über IM-Protokolle, wie z. B. das Jabber-Protokoll [67], denkbar.

Man geht davon aus, dass dieser Dienst (IM und auch VoIP) für die Kommunikation notwendig ist und sich jeder Teilnehmer sofort nach dem Anschluss ans Netzwerk daran anmelden muss. Deshalb versucht man, eine doppelte Authentifizierung (Netzzugang und Applikation) zu vermeiden und die Authentifizierung am entsprechenden Dienst für die Netzwerkautorisation mit zu nutzen.

Auch die Authentifizierung über ein IM-Protokoll unterliegt ebenfalls den in Abschnitt 6.3.2.1 beschriebenen Problemen.

6.3.2.3 Zusammenfassung

Für die Authentifizierung und Autorisation im WLAN bieten sich dieses Verfahren nicht an. Da es hierfür keine standardisierten Verfahren gibt, müssten diese selbst erstellt und gepflegt werden. Ein sinnvoller Einsatz dieser Verfahren ist lediglich bei Geräten gegeben, die ausschließlich diese Anwendungen zur Verfügung stellen.

7 Implementierung

7.1 Voraussetzungen

Für das bessere Verständnis der in diesem Kapitel beschriebenen Implementierungen sei hiermit noch einmal auf die im Kapitel 2 behandelten Hintergründe hingewiesen.

Die nachfolgenden Abschnitte geben einen Einblick in die bestehenden Verwaltungssysteme des CSN, die für die Implementierungen der WLAN-Infrastruktur modifiziert werden müssen.

7.1.1 Netzstruktur

Der Aufbau des CSN-Netzwerkes ist im Abschnitt 2.1.2 abgebildet. Ausgehend davon wird diese Implementierung als Pilotprojekt im Wohnheim Vettersstraße 52 integriert. Dazu müssen für das WLAN-Projekt an den Netzkomponenten dieses Hauses entsprechende Konfigurationen vorgenommen werden.

7.1.2 Firewallsystem

Das Firewallsystem des CSN wird auf dem CSN-Server (siehe Anhang 9.3.1.1) betrieben. Dabei bildet dieser die Schnittstelle zwischen dem Netzwerk des CSN und dem Campusnetz der TU Chemnitz. Über diesen Anschluss ist die Anbindung an das Internet realisiert.

Das Firewallsystem basiert auf einem Regelsatz, der mittels Netfilter [68] definiert wird. Hierbei gibt es einen generellen Regelsatz, der für alle IPs aus den Subnetzen des CSN gilt.

Danach existieren generelle Regeln für verschiedene Netzklassen, die im Rahmen der Studienarbeit „*Klassenbasierter Anschluss im Chemnitzer Studenten Netz*” [69] von Sebastian Junge umgesetzt wurden.

Für jede vergebene IP aus den Subnetzen des CSN werden Regeln in das Firewallsystem eingetragen, die eine spezielle Markierung setzen. Anhand dieser Markierung wird entschieden, zu welcher Netzklasse sie gehören und somit welcher Regelsatz angewendet werden muss. Weiterhin kann das System damit feststellen, ob es sich um Datenverkehr von/nach außerhalb der Subnetze der TU Chemnitz handelt und welchem Nutzer dieser Datenverkehr zugeordnet wird.

Eine ausführliche Erklärung dieses Konzeptes wurde in einem Dokument mit dem Titel „*A Packet filtering Concept based on iptables MARK's* ” [70] von Markus Schade und Daniel Schreiber veröffentlicht.

7.1.3 Dynamisches Bandbreitenmanagement (Shaper)

Da bereits das Firewallsystem den Datenverkehr entsprechend markiert, kann das Shapersystem diese Markierungen nutzen. Als Erstes wird entschieden welcher Datenverkehr eingeschränkt werden muss. Dies ist im Falle des CSN der Datenverkehr von und nach außerhalb des Subnetzbereiches der TU Chemnitz. Nur für diesen sogenannten „externen Traffic” wird die Bandbreitenbeschränkung angewandt. Weiterhin wird anhand der Markierung die für den Nutzer ermittelte Bandbreite für diese Verbindungen verwendet.

Weiterführende Informationen zum Shapingsystem sind bereits im Abschnitt 2.1.4 enthalten.

7.1.4 Traffic-Accounting

Wie bereits im vorangegangenen Abschnitt beschrieben muss das Datenvolumen des „externen Traffics” erfasst werden. Dies muss aus den oben genannten Gründen pro IP bzw. pro Nutzer aufgezeichnet und analysiert werden.

Um einer übermäßigen Belastung des Netzwerkes entgegenzuwirken, wurde eine Grenze für das gesamte Datenvolumen pro Nutzer und Tag festgelegt. Deshalb wird dieses ebenfalls erfasst.

Für die Erfassung und Einordnung des Datenvolumens werden im CSN die durch das Cisco Netflow [71] am zentralen Router (siehe Anhang 9.3.2.1) ermittelten Werte gespeichert und analysiert. Dies geschieht am Accounting-Server (siehe Anhang 9.3.1.2) durch die Flow-Tools [72] und wird viertelstündlich durch entsprechende Programme in das Datenbanksystem des CSN transferiert. Hierbei wird das Datenvolumen durch entsprechende Filterregeln pro IP wie oben beschrieben kategorisiert und aufsummiert. Eine Zuordnung der Trafficwerte pro IP zu einem Nutzer geschieht durch die in der Datenbank gespeicherten Relationen. Es ist somit möglich, das Datenvolumen für den „externen Traffic” (Up- und Download) sowie den Gesamttraffic (Up- und Download) pro Nutzer über bestimmte Zeiträume zu ermitteln und z. B. für die Steuerung des Shapingsystems zu nutzen.

7.1.5 Netzwerksicherheit

Gegen mögliche Manipulationen und zur Zugangskontrolle gegen unberechtigte Nutzung des Netzwerkes wurden verschiedene Sicherheitsmechanismen aktiviert. Hierzu zählt die sogenannte Port-Security, die nur festgelegte Hardwareadressen der Netzwerkkarten an den Ports der Netzwerkkomponenten zulässt. Der Port wird durch entsprechende Programme anhand des aktuellen Datenbankzustandes konfiguriert. Daraus wird ebenfalls das sogenannte „Static ARP” am zentralen Router angepasst, das Änderungen der zugeordneten IP-Adresse verhindert. Somit existiert eine feste Zuordnung von IP- und Hardware-Adresse, die wiederum dem Nutzer zugeordnet werden kann.

7.1.6 Nutzer-Authentifizierung

Eine Authentifizierung des Nutzers ist bisher nur für die Webseiten des Verwaltungssystems notwendig. Dies wird über HTTP-Authentifizierung realisiert. Dazu wird auf die notwendigen Daten des Datenbestandes der TU Chemnitz zurückgegriffen. Eine explizite Anmeldung des Nutzers ist deshalb nicht notwendig.

Zu diesem Zweck werden in regelmäßigen Abständen die Logins und verschlüsselten Passwörter der Studenten vom Verwaltungssystem des URZ in das des CSN transferiert.

7.2 Entwurfsentscheidung

7.2.1 WLAN-Architektur

Der WLAN-Standard 802.11b/g ist in Europa am meisten verbreitet, deshalb werden dafür die meisten Netzwerkkarten angeboten/verkauft. Meistens sind diese Karten bereits in aktuellen Laptops und anderen mobilen Geräten integriert. Eher selten ist dagegen der 802.11a Standard zu finden. Da der größere Nutzen beim b/g-Standard liegt, wurde dieser in der praktischen Umsetzung verwendet.

Für die Netzstruktur wurden deshalb zwei Access Points für den 802.11b/g Standard erworben (siehe Anhang 9.3.2.4). Dazu gehören auch die Antennenkabel und die Outdoor-Antennen.

7.2.2 WLAN-Architektur

Für die Erweiterung des bestehenden Netzwerkes des CSN sind größtmögliche Sicherheit und Skalierbarkeit zu gewährleisten. Dabei ist aber eine einfache Handhabung und eine gute Unterstützung der am weitesten verbreiteten Betriebssysteme zu beachten. Aus diesem Grund wurde sich für die Realisierung des 802.11i Standards entschieden. Hierbei wird die Verbindung mittels TKIP oder CCMP gesichert und eine nutzerbasierende Authentifizierung mittels 802.1x realisiert.

Aber auch Systemen, für die keine 802.1x Unterstützung existiert, soll die Möglichkeit geboten werden, die WLAN-Infrastruktur zu nutzen. Hierfür wird ein „offenes WLAN” ohne Authentifizierung (siehe Open Systen Authentication (siehe Abschnitt 4.3.1.3)) und Verschlüsselung auf der Sicherungsschicht realisiert, da WEP durch seine bekannten Schwächen (siehe Abschnitt 4.3.1.4) keinen wirklichen Schutz bietet. Für eine sichere Verbindung mit Authentifizierung und Verschlüsselung ist die Anwendung eines VPN (siehe Abschnitt 4.4) notwendig. Das „offene WLAN” bietet dabei lediglich den Zugang zum VPN-Server, andere Datenverbindungen werden blockiert.

7.2.3 Netzintegration

Die verwendeten Access Points bieten dabei die Möglichkeit mehrere verschiedene WLANs parallel zu betreiben. Daher werden die beiden Access Points als ESS betrieben und bieten beide im vorigem Kapitel beschriebenen Zugangsmöglichkeiten parallel an. Durch eine entsprechende Montage der Geräte und Antennen decken sie damit ein größeres Gebiet ab.

Der Datentransport ins drahtgebundene Netzwerk geschieht mittels verschiedener VLANs (Virtual LAN) und kann mittels VLAN-Trunk (nach IEEE 802.1q [73]) über den Ethernet-Port des Access Points an die nächste Netzkomponente übertragen werden.

Für die Konfiguration, Überwachung und NAS-Funktionalität wird den Access Points jeweils eine IP-Adresse zugewiesen. Diese sind aus dem Subnetzbereich des Management-Netzes, das nur von den CSN-Servern erreichbar und einem weiteren VLAN zugeordnet ist.

Angeschlossen sind die beiden Geräte jeweils an einem Switch, der für den Weitertransport der Daten sorgt. Die entsprechenden VLANs sind dafür an den Anschlüssen konfiguriert.

7.2.3.1 WLAN mit 802.11i

Für dieses WLAN wurde die SSID **CSN_WPA** definiert und dem *VLAN 90* zugeordnet. Der Zugang zu diesem WLAN ist erst nach erfolgreicher Authentifizierung mittels 802.1x möglich. Für die Datenverschlüsselung bieten die Access Points sowohl TKIP als auch CCMP an. Je nach Funktionalität der WLAN-NIC der mobilen Station wird eines der beiden Verfahren angewandt. Dabei erfolgt der Schlüsselaustausch über 802.1x, wie in Abschnitt 4.3.2.3 beschrieben.

Sobald die *Association* abgeschlossen ist, kann der WLAN-Client eine IP-Adresse mittels DHCP beziehen. Diese wird dynamisch durch den DHCP-Server auf dem CSN-Server vergeben und ist Teil des offiziellen CSN-IP-Adressraums. Somit besitzt der WLAN-Client die gleiche Netzverbindung wie alle anderen drahtgebundenen CSN-Anschlüsse.

Folgende Abbildung zeigt die Integration ins bestehende Netz:

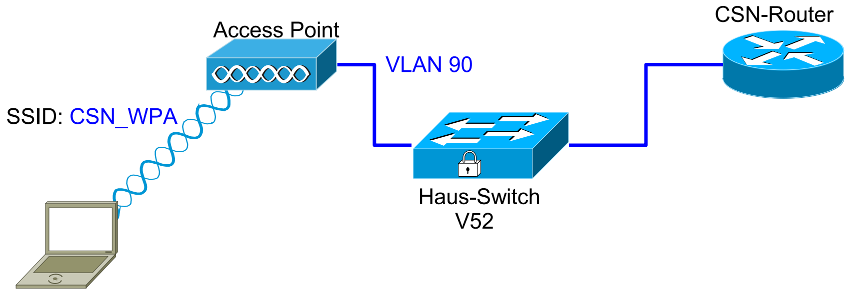

Abbildung 7.1: CSN-WLAN mit 802.11i

Damit besitzt dieses WLAN folgende Eigenschaften:

- **SSID:** CSN_WPA
- **Authentifizierung:** 802.1x (EAP)
- **Verschlüsselung:** TKIP/AES CCMP
- **IP-Vergabe:** dynamisch
- **IP-Adressen:** 134.109.90.0/255.255.254.0

7.2.3.2 Offenes WLAN mit VPN-Zugang

Anders als das eben beschriebene WLAN handelt es sich hierbei um ein „offenes WLAN”. Es setzt keine Authentifizierung zur *Association* voraus. Diesem WLAN wurde die SSID **CSN** gegeben und dem *VLAN 10* zugewiesen. Es ist keine Verschlüsselung für die Sicherungsschicht vorgesehen.

Es werden IP-Adressen dynamisch nach RFC 1918 [74] für die Benutzung in „privaten Netzwerken” vergeben, die aber nicht in das CSN-Netzwerk weitergeleitet werden. Der für dieses Subnetz zuständige DHCP-Server befindet sich auf einem extra Server (siehe Anhang 9.3.1.3) und regelt die im folgenden beschriebenen Zugangsbeschränkungen.

Damit besitzt dieses WLAN folgende Eigenschaften:

- **SSID:** CSN
- **Authentifizierung:** keine
- **Verschlüsselung:** keine
- **IP-Vergabe:** dynamisch
- **IP-Adressen:** 172.16.10.0/255.255.254.0

Zugangsbeschränkung Das „offene WLAN” soll lediglich für die Verbindung zum VPN-Server nutzbar sein. Um jedoch den Nutzern die Konfigurationsbeschreibung und die notwendige Software zugänglich zu machen, wurde ein extra Server (siehe Anhang 9.3.1.3) für die Zugangssteuerung installiert. Dabei ist er für das Netzwerk das „Default-Gateway”, dient als „transparenter Proxy” und lenkt alle durch den Nutzer im Browser eingegebenen URLs (Uniform Resource Locator) auf die Webseite mit den Informationen zum WLAN-Zugang [75] um. Alle anderen Verbindungen werden durch das System geblockt. Informationen zur installierten Software und dessen Konfiguration befinden sich im Anhang 9.5.2.

VPN-Zugang Da für dieses Netzwerk Authentifizierung und Verschlüsselung fehlen, erwies sich der Einsatz eines VPN (siehe Abschnitt 4.4) für den gesicherten Datenaustausch als sinnvoll bzw. notwendig. Auf IPSec (siehe Abschnitt 4.4.1) konnte aus verschiedenen Gründen nicht aufgesetzt werden: So fehlt dem Protokoll die Authentifizierungsinstanz, der Implementierungsstand in den Betriebssystemen ist sehr unterschiedlich und die Konfiguration meist komplex und aufwendig. Die Protokolle PPTP und L2TP werden wegen mangelnder Sicherheit nicht eingesetzt.

Die Entscheidung fiel auf die proprietäre VPN-Lösung von Cisco Systems. Entscheidend waren dabei die große Anzahl an unterstützten Betriebssystemen, der einfach zu bedienende Client und die Verschlüsselung mittels IPSec. Weiterhin bietet dieses System zusätzliche Zugangsmöglichkeiten, wie z. B. einen SSL-VPN-Client usw.

Unterstützte Betriebssysteme:

- Microsoft Windows 9x/ME/XP/2000
- Linux ab Kernel 2.2 (für x86 und x86_64)
- MacOS X
- Solaris
- PalmOS/Windows Mobile 2000 (über Drittanbieter)

Dazu hat der VPN-Server, auch *VPN-Concentrator* (siehe Anhang 9.3.2.2) genannt, eine IP-Adresse aus dem Subnetz des „offenen WLAN" und bildet den Zugang zum CSN-Netzwerk.

Die VPN-Software befindet sich bereits vorkonfiguriert zum Download auf dem Webserver des CSN. Nach der Installation und Aktivierung der Software muss sich der Nutzer nur noch mit seinem Login und Passwort authentifizieren. Nach erfolgreicher Authentifizierung am *VPN-Concentrator* (siehe Kapitel 7.3.4) wird ein *Tunnel* zwischen ihm und dem WLAN-Client aufgebaut, durch den der gesamte Datenverkehr geleitet wird. Der *Tunnel* hat eine offizielle IP-Adresse aus dem CSN-IP-Adressraum und bietet damit den gleichen Zugang zum CSN-Netzwerk wie alle anderen drahtgebundenen CSN-Anschlüsse.

Folgende Abbildung zeigt die Integration des WLANs mit VPN-Zugang:

Damit besitzt das WLAN mit VPN folgende Eigenschaften:

- **SSID:** CSN
- **Authentifizierung:** mittels VPN-Software
- **Verschlüsselung:** mittels VPN-Software
- **IP-Vergabe:** dynamisch durch VPN-Concentrator
- **IP-Adressen:** 134.109.118.0/255.255.254.0

7.3 Authentifizierung und Autorisation

Bisher beschränkt sich die Authentifizierung bzw. Autorisation im CSN auf:

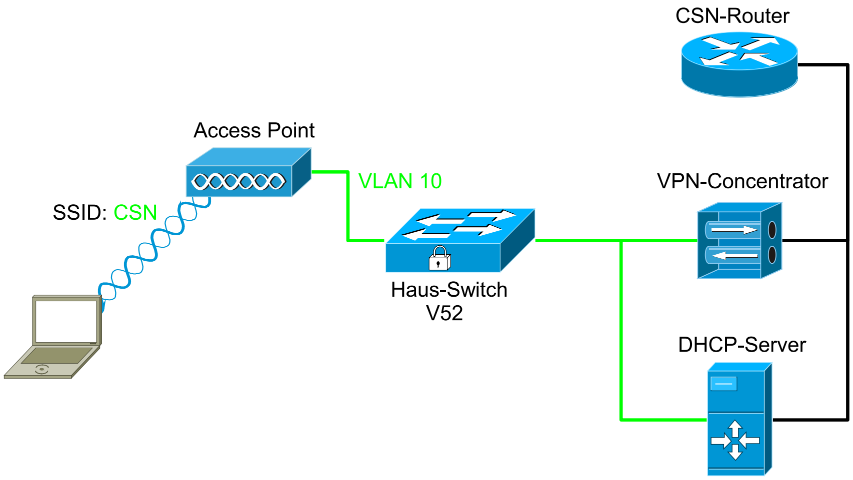

Abbildung 7.2: CSN-WLAN mit VPN

- MAC-Adresse der Netzwerkkarte für den Zugang zum Netzwerk
- Login und Passwort für den Zugang zu den Webseiten des CSN mittels HTTP-Authentifizierung nach RFC-2617 [76]

Ziel dieser Implementierung ist es, ohne zusätzliche Anmeldung des mobilen Geräts den WLAN-Zugang nutzen zu können. Aus diesem Grund ist die Authentifizierung anhand der Hardware-Adresse in diesem Fall nicht möglich. Wie bereits schon beschrieben, ist dieses Verfahren der Zugangskontrolle im WLAN nicht sinnvoll und sicher anwendbar.

Die Möglichkeit der Authentifizierung und Autorisation für den Netzzugang mittels HTTP/HTTPS wurde bereits im Abschnitt 6.3.1 beschrieben. Es ist bereits ausführlich untersucht worden und wird deshalb an dieser Stelle nicht weiter vertieft.

Um die WLAN-Zugänge anhand einer nutzerbasierenden Authentifikation zu realisieren, bietet sich eine AAA-Umgebung an.

7.3.1 AAA-Protokoll

Im Kapitel 6 wurden bereits die Protokolle RADIUS, DIAMETER und TACACS näher untersucht und stehen deshalb hier zur Auswahl.

TACACS ist ein veraltetes Protokoll, das hauptsächlich durch TACACS+ abgelöst wurde, das wiederum überwiegend in Netzkomponenten von Cisco Systems zu finden ist. Freie Serverimplementierungen werden zum Teil seit Jahren nicht mehr gepflegt und weiterentwickelt.

DIAMETER ist momentan noch in der Entwicklungsphase, wodurch freie Serverimplementierungen noch fehlen bzw. bisher nur in Teilen implementiert wurden und so nicht die notwendigen Funktionen zur

Verfügung stehen. Weiterhin fehlt noch immer die DIAMETER-Unterstützung in den Netzkomponenten. Auch wenn DIAMETER mit seinen Eigenschaften, wie sicherer Datentransport, Session-Beendung durch Server usw., die gewünschte Funktionalität bietet, ist es momentan leider noch nicht einsatzbereit.

RADIUS hat sich seit Jahren als AAA-Protokoll etabliert und wird in vielen Netzkomponenten unterstützt. Außerdem gibt es viele freie und kommerzielle Implementierungen, die auch aktuell noch gepflegt, überarbeitet und erweitert werden. Damit bietet sich die Nutzung des RADIUS-Protokolls an.

7.3.1.1 FreeRADIUS

Eine der größten und aktivsten freien RADIUS-Implementierungen ist das FreeRADIUS-Projekt [47]. Einige der wichtigsten Funktionalitäten des FreeRADIUS sind:

- EAP (EAP-MD5, EAP-SIM, EAP-TLS, EAP-TTLS, EAP-PEAP, Cisco LEAP)
- Datenbank-Unterstützung von: MySQL, PostgreSQL, Oracle, LDAP
- unterstützt Proxyfunktionalität, Fail-Over und Load-Balancing
- für Linux, BSD, ...

7.3.1.2 FreeRADIUS und Debian

Wie aus Anhang 9.3.1.1 ersichtlich, werden die Server im CSN mit der Debian-Distribution [77] betrieben. Dabei wird auf die *stable* Version „Sarge” gesetzt. Das hier verfügbare Debian-*Package* von FreeRADIUS muss aus rechtlichen Gründen ohne EAP-Support und ohne PostgreSQL-Unterstützung auskommen. Beides wird aber für den geplanten Betrieb des Systems benötigt, EAP für das WLAN mit 802.11i und die PostgreSQL-Unterstützung, um auf die CSN-Datenbank zugreifen zu können.

Deshalb wurde dieses *Package* mit der kompletten Unterstützung neu zusammengestellt und installiert. Dabei ist darauf zu achten, dass Debian in der aktuellen Version die `libtools` modifiziert hat und bei `libltdl3` auf die CVS (Concurrent Versions System)-Version setzt. Dadurch gibt es weitere Probleme, die nur durch einen Patch behebbar sind. Die aktuell installierte Version von FreeRADIUS ist also eine modifizierte Version `1.0.2`.

FreeRADIUS wird als Systemdienst gestartet und setzt dabei als Datenbasis auf das Datenbanksystem PostgreSQL mit der CSN-Datenbank. Dabei wurden einige Veränderungen vorgenommen, die im weiteren Verlauf näher erläutert werden. Die Konfigurationsdateien von FreeRADIUS und die Anpassungen der Datenbank befinden sich im Anhang 9.4.

7.3.1.3 FreeRADIUS Integration

Die Konfiguration des FreeRADIUS wird durch die Konfigurationsdatei `radiusd.conf` durchgeführt. Hier wird die Funktionalität des Servers eingestellt. Dabei kann der FreeRADIUS die Nutzerdaten aus Dateien oder Datenbanken beziehen.

Da der RADIUS-Server auf die CSN-Datenbank zurückgreifen soll muss der Parameter „*sql*” in der `radiusd.conf` in den Konfigurationabschnitten *authorize*, *session* und *post-auth* eingesetzt werden. Parameter, die das Benutzen von Dateien konfigurieren, können entfernt werden.

Weiterhin müssen noch die Dateien `eap.conf` und `postgresql.conf` angepasst und aktiviert werden. Die entscheidenden Konfigurationsabschnitte sind im Anhang 9.5.1 zu finden. Für ausführliche Informationen zur Konfiguration des FreeRADIUS sei hiermit auf dessen Dokumentation [**?**] verwiesen.

7.3.2 FreeRADIUS und CSN-Datenbank

FreeRADIUS erwartet die Daten aus der Datenbank in verschiedenen Datenbank-Tabellen und -Spalten. Um die redundante Speicherung von Daten zu vermeiden, gibt es bei PostgreSQL sogenannte „Views”, auch „Sichten” genannt. Dies sind gespeicherte SELECT-Abfragen durch die sich gespeicherte Daten in geforderten Reihenfolgen/Formaten darstellen lassen.

Von dieser Art und Weise Daten aufzubereiten wird nachfolgend Gebrauch gemacht, da sich viele notwendige Daten bereits in anderer Form in der CSN Datenbank befinden. Die folgenden Abschnitte beschreiben kurz die für den FreeRADIUS angelegten Views. Die Definitionen dieser sind im Anhang 9.4 abgebildet.

7.3.2.1 Konfiguration für NAS

Im Datenbestand des CSN sind alle Netzkomponenten verzeichnet. Aufbauend auf diese Daten müssen dem FreeRADIUS alle NAS-Geräte übermittelt werden, von denen aus AAA-Anfragen erwartet werden. Dabei existiert für die Authentifizierung des NAS gegenüber dem RADIUS-Server ein gemeinsames Geheimwort, das als neuer Wert für die entsprechende Netzkomponente in der entsprechenden Gerätetabelle (Tabelle: *hub*) gespeichert wird. Da diesen Wert nur die NAS-Geräte besitzen, wird diese für die Erstellung der View (View: *nas*) genutzt.

7.3.2.2 Nutzer-Authentifizierung/-Autorisation

Für die Nutzer-Authentifizierung/-Autorisation sind folgende Datenbank-Tabellen im FreeRADIUS definiert worden: `radcheck`, `radusergroup`, `radgroupcheck` und `radgroupreply`

Dabei beinhaltet `radcheck` Logins und Passwörter. Das Attribute *Crypt-Password* weist den FreeRADIUS an, dass das im *value*-Datenfeld stehende Passwort verschlüsselt ist.

In der Tabelle `radusergroup` befindet sich die Zuordnung von Login zur zugehörigen Gruppe. Für eine Veranschaulichung der Nutzung dieser Gruppen wurden zwei davon angelegt, eine „Admin”- und eine „Nutzer”-Gruppe. Die „Admin”-Gruppe setzt sich dabei aus den CSN-Teammitgliedern zusammen, alle anderen gültigen Logins werden der Gruppe „Nutzer” zugeordnet.

Die Tabelle `radgroupcheck` beinhaltet die nutzergruppenspezifischen Attribute, die während der Authentifizierung überprüft wurden. Hier wurde beispielhaft anhand der Nutzergruppen die maximale Anzahl paralleler Sessions festgelegt. Dazu dient das Attribute *Simultaneous-Use*. Für die Nutzergruppe „Admin” wurde der Wert auf 10 gesetzt, während der der Nutzergruppe „Nutzer” auf 1 steht. Falls die angegebene Anzahl bereits erreicht ist, wird die Authentifizierung abgelehnt.

In der letzten Tabelle `radgroupreply` befinden sich nutzergruppenspezifische Attribute, die bei erfolgreicher Authentifizierung im *Response* dem NAS übergeben werden. Hier wird eine maximale Session-Zeit eingestellt. Nach Ablauf dieser Zeit trennt der NAS den Client und der Nutzer muss sich neu anmelden und authentifizieren. Hierzu dient das Attribute *Session-Timeout*, das für die Nutzergruppe „Nutzer” auf 86400 Sekunden eingestellt ist, was 24 Stunden darstellt. Bei der Nutzergruppe „Admin” wurde bei diesem Attribut kein Wert gesetzt, womit es keine Session-Zeit-Begrenzung gibt.

Diese vier vom FreeRADIUS benötigten bzw. erwarteten Tabellen wurden ebenfalls als *View* realisiert. Die Erzeugungsvorschrift und beispielhafte Darstellungen befinden sich im Anhang 9.4.1.

Eine weitere Möglichkeit, nach erfolgreicher Authentifizierung dem NAS bestimmte Attribute durch den RADIUS-Server mitzugeben, ist mittels der Werte in der Tabelle `radreply`. Hier können nutzerspezifische Attribute gesetzt werden. Darauf wurde hier aber verzichtet.

7.3.3 WLAN mit 802.11i

7.3.3.1 Access Point Konfiguration

Für die Authentifizierung/Autorisation muss man am Access Point für die entsprechende SSID (*CSN_WPA*) die Authentifizierungsmethode EAP einstellen und den entsprechenden RADIUS-Server eintragen, in diesem Fall den CSN-Server auf dem der FreeRADIUS läuft. Dabei muss auch das gemeinsame Geheimwort angegeben werden, was vorher in der Gerätetabelle gespeichert wurde.

Der entsprechende Teil der Konfiguration des Access Points ist im Anhang 9.5.3.1 zu finden.

7.3.3.2 WLAN-Client-Konfiguration

Damit sich der WLAN-Client an das Netz anmelden kann, muss das WLAN-Interface so eingestellt werden, dass es sich mittels 802.1x authentifiziert. Dazu muss hierbei **EAP-TTLS** mit **PAP**-Authentifizierung genutzt werden. Der Grund hierfür ist, dass im Datenbestand des CSN lediglich verschlüsselte Passwörter gespeichert sind. Login und Passwort werden bei **PAP** im Klartext und dabei mittels **EAP-TTLS** verschlüsselt übertragen. Der RADIUS-Server kodiert das empfangene Passwort und vergleicht es zusammen mit dem Login mit seinem Datenbestand (View: `radcheck`).

Nicht in jedem Betriebssystem ist **EAP-TTLS** bereits unterstützt, aber in den meisten Fällen durch Installation von Zusatzsoftware damit erweiterbar:

- **Windows XP/2000** mittels *SecureW2* [78]
- **Linux** mittels *wpa_supplicant* [36]
- **MacOS X** bereits im Betriebssystem integriert
- **Pocket PC 2003** mittels *SecureW2* [78]
- **Windows Mobile 2003 SE** mittels *SecureW2* [78]

Die Software und die Konfigurationsanleitung wurde den Nutzern auf der entsprechenden Webseite [75] auf dem CSN-Server zur Verfügung gestellt.

7.3.3.3 FreeRADIUS Konfiguration

Beim FreeRADIUS muss die EAP Unterstützung aktiviert sein. Dazu müssen das Server- und das CA-SSL-Zertifikat installiert und in der `eap.conf` entsprechend angegeben sein. Im Anhang 9.5.1.3 ist die installierte Konfiguration abgebildet.

7.3.4 Offenes WLAN mit VPN

Hierfür muss am Access Point lediglich die SSID und das entsprechende VLAN konfiguriert sein. Der dafür relevante Konfigurationsteil der Access Points befindet sich im Anhang 9.5.3.2.

7.3.4.1 VPN-Concentrator

Da der VPN-Concentator den Authentifikator darstellt, muss ihm der RADIUS-Server bekanntgegeben werden, an den er die Authentifizierungsanfragen weiterleitet. Dafür muss auf diesem Gerät ein Nutzergruppe definiert werden, die alle für die Nutzung notwendigen Einstellungen, wie IP-Adressraum, Authentifizierungsserver usw., enthält. Weiterhin werden dafür eine entsprechendes Gruppen-Login und -Passwort gesetzt, die für die Verschlüsselung der Datenverbindung notwendig sind. Eines der drei Netzwerkinterfaces ist so konfiguriert, das es zum Subnetz des „offenen WLANs" (VLAN 10) gehört. Nur von diesem aus sind VPN-Verbindungen möglich.

Die wichtigste Einstellung für die angelegte Nutzergruppe des VPN-Concentrators ist die des Authentifizierungsservers. Hierbei wurde der CSN-Server, auf dem der FreeRADIUS läuft, auf den entsprechenden TCP/IP Port 1813 und das entsprechende gemeinsame Geheimwort eingestellt.

Der VPN-Client transportiert die Login/Passwort-Kombination im Klartext über die verschlüsselte Verbindung zum VPN-Concentrator. Dieser leitet diese Kombination entsprechend im RADIUS-Protokoll mittels PAP an den Authentifizierungsserver weiter, der somit auch mit den in der Datenbank gespeicherten verschlüsselten Nutzerpasswörtern arbeiten kann.

Die Konfiguration des VPN-Concentrators geschieht ausschließlich über dessen Webinterface. Dem interessierten Leser sei dazu die Konfigurationsdokumentation [79] des Gerätes empfohlen.

7.3.4.2 VPN-Client

Da der VPN-Client von Cisco Systems den geringsten Supportaufwand hat, wurde sich dafür entschieden. In dieser Software kann man bereits die passende Konfiguration mit ausliefern, die bei der Installation sofort aktiviert wird. Nach der Installation und dem Start der Client-Software kann der Nutzer das entsprechende *CSN-Profil* auswählen und die Verbindung starten. Nach Eingabe der Login/Passwort-Kombination wird der VPN-Tunnel aufgebaut. Neben dem *CSN-Profil* konnte auch das der TU Chemnitz mit dazu gegeben werden, womit der VPN-Client auch für die VPN-Zugänge auf dem Campusnetz ohne Änderungen nutzbar ist.

Der VPN-Client von Cisco Systems ist für folgende Betriebssysteme auf den CSN-Webseiten verfügbar und darf von allen angemeldeten Nutzern kostenfrei genutzt werden:

- **Windows 9x/ME/XP/2000**
- **Linux**
- **MacOS X**

Für die Betriebssysteme *Pocket PC 2003* und *Windows Mobile 2003 SE* wird von anderen Herstellern eine passende Software angeboten, die allerdings vom CSN nicht kostenfrei angeboten werden kann. Für Linux existiert zusätzlich eine Open Source Software (`vpnc` [80]), die allerdings nicht alle Verbindungsarten unterstützt und manuell konfiguriert werden muss.

7.3.4.3 FreeRADIUS Konfiguration

Der VPN-Concentrator setzt auf das RADIUS-Protokoll mit PAP-Authentifizierung, deshalb müssen für die Nutzung am FreeRADIUS keine weiteren Einstellungen vorgenommen werden.

7.4 Accounting

7.4.1 Allgemein

Für die Abrechnungsbarkeit der Sitzungen ist es wichtig, dass diese aufgezeichnet werden. Dafür ist in den AAA-Protokollen das *Accounting* vorgesehen. Nach erfolgreicher Authentifizierung setzen die NAS, also sowohl die Access Points als auch der VPN-Concentrator, einen *Accounting-Start-Request* an den RADIUS-Server ab. Sobald der Client vom NAS getrennt wird, wird ein *Accounting-Stop-Request* gesendet. Hierbei werden jeweils Datum und Uhrzeit durch den Server festgehalten. Weiterhin werden Werte wie z. B. SessionID, NAS-IP-Adresse und -Type sowie das Nutzerkennzeichen zur Sitzung archiviert.

Für die Abrechnung des Datenaufkommens während der Session muss zusätzlich zum Nutzerlogin die zugewiesene dynamische IP-Adresse gespeichert werden. Dies ist notwendig um das entstandene Datenvolumen, wie unter Abschnitt 7.1.4 beschrieben, zu erfassen. Dies geschieht für den VPN- und 802.11i-Zugang auf unterschiedliche Art und Weise. Die genaue Erklärung folgt in den nächsten Abschnitten.

Das *Accounting* dient hierbei nicht nur der Aufzeichnung der verschiedenen Sitzungsdaten, sondern auch zur vollständigen (De-)Aktivierung des CSN-Zugangs. Dazu wurde eine entsprechende Erweiterung des *Accounting*-Dienstes implementiert. Hierbei werden die zugewiesene IP-Adresse der Sitzung und der zugehörige Benutzer erfasst und dies passend zum Firewallkonzept (siehe Abschnitt 7.1.2) des CSN am CSN-Server aktiviert.

Dazu bietet der FreeRADIUS die Möglichkeit beim *Accounting-Start* und *Accounting-Stop*-Request bestimmte Programme auszuführen. Die entsprechende Konfigurationsdatei `acct_users` ist im Anhang 9.5.1.4 abgebildet. Die Funktionsweise wird in den nächsten Abschnitten genauer erklärt.

7.4.2 FreeRADIUS Konfiguration

Die Sitzungsinformationen sollen durch den FreeRADIUS ebenfalls in der Datenbank abgespeichert werden. Dazu muss in der `radiusd.conf` im Abschnitt *accounting* der Parameter *sql* eingefügt werden, andere Parameter können entfernt werden. Damit der FreeRADIUS die Konfigurationsdatei `acct_users` einliest und somit die definierten Programme ausführt, muss in der `radiusd.conf` unter Abschnitt *preacct* der Parameter *files* enthalten sein.

In der Konfigurationsdatei für das Datenbank-Backend `postgresql.conf` ist für das Accounting die Tabelle *radacct* vorgesehen. Diese wurde unverändert, wie vom FreeRADIUS definiert, in der CSN-Datenbank angelegt. Zusätzlich wurde daran ein „Trigger" gebunden, der beim INSERT oder UPDATE auf diese Tabelle die Werte für *acctstarttime* und *acctstoptime* auf die vorherige bzw. nachfolgende Viertelstunde rundet. Dies ist für das Trafficaccounting notwendig, da hierbei die Datenwerte viertelstündlich eingetragen werden. Die Einträge in dieser Tabelle werden 70 Tage archiviert und durch eine Datenbankfunktion beim Überschreiten dieser Zeit, zusammen mit anderen Daten, wie z. B. Trafficwerten, aus der Tabelle gelöscht.

Die Definition der *Accounting*-Tabelle und des „Trigger" sind im Anhang 9.4.2 abgebildet.

7.4.3 VPN-Concentrator

Beim *Accounting-Start-Request* durch den VPN-Concentrator werden u. a. folgende Informationen gespeichert (Bezeichner der Tabellenspalte in Kammern):

- Nutzerlogin (*username*)
- aktuelles Datum und Zeit (*acctstarttime*)
- NAS-IP-Adresse (*nasipaddress*)
- NAS-Type (*nasporttype*)
- Tunnel-IP-Adresse (*framedipaddress*)
- Client-IP-Adresse (*callingstationid*)

Beim *Accounting-Stop-Request* werden zusätzliche Daten aufgezeichnet:

- aktuelles Datum und Zeit (*acctstopttime*)
- Verbindungsdauer (*acctsessiontime*)
- Trafficdaten (*acctinputoctets* und *acctoutputoctets*)

Der VPN-Concentrator schickt nur zum Sitzungsbeginn und -ende die entsprechenden Accounting-Werte. Alle notwendigen Informationen für die Konfiguration der Firewall und der CSN-Trafficauswertung sind bereits in den Accounting-Daten enthalten.

7.4.4 WLAN mit 802.11i

Beim *Accounting-Start-Request* durch den jeweiligen Access Point werden u. a. folgende Informationen gespeichert:

- Nutzerlogin (*username*)
- aktuelles Datum und Zeit (*acctstarttime*)
- NAS-IP-Adresse (*nasipaddress*)
- NAS-Type (*nasporttype*)
- Client-MAC-Adresse (*callingstationid*)

Beim *Accounting-Stop-Request* werden zusätzliche Daten aufgezeichnet:

- aktuelles Datum + Zeit (*acctstopttime*)
- Verbindungsdauer (*acctsessiontime*)
- Trafficdaten (*acctinputoctets* und *acctoutputoctets*)

In regelmäßigen, konfigurierbaren Abständen übersendet der jeweilige Access Point *Accounting-Interim-Records* zur aktiven Sitzung. Darin sind z. B. die aktuellen Trafficdaten enthalten.

Für die Konfiguration der CSN-Firewall und der CSN-Trafficauswertung fehlt die, dem WLAN-Client zugewiesene, IP-Adresse. Diese kann aus den eingesetzten Access Points nach dem *Accounting-Start-Request* mittels SNMP (Simple Network Management Protocol) ausgelesen werden. Weitere Informationen, die zusätzlich in den Access Points gespeichert sind, aber nicht in den *Accounting*-Requests übermittelt werden, können ebenso ausgelesen werden.

7.4.5 FreeRADIUS-Accounting-Erweiterungen

Wie bereits beschrieben, müssen zu Beginn und am Ende einer Sitzung nicht nur Daten in die Accounting-Tabelle geschrieben werden, sondern auch entsprechende Regeln in der CSN-Firewall aktiviert werden.

Die Gründe dafür sind:

- **Sicherheit** - Nur die verwendeten IP-Adressen werden in die CSN-Firewall eingetragen. IP-Adressen die nicht enthalten sind, werden geblockt. Somit kann kein Datenverkehr von nicht ordnungsgemäß vergebenen IP-Adressen das CSN verlassen.
- **Bandbreitenbeschränkung** - In der CSN-Firewall werden den IP-Adressen nutzerspezifische Markierungen zugewiesen, anhand dieser der Shaper die Bandbreite bestimmt.

Für alle statisch vergebenen IP-Adressen aus dem drahtgebundenen Netzwerk übernehmen selbstgeschriebene Programme die Konfiguration zu jeder vollen Stunde. Da WLAN- bzw. VPN-Sitzungen aber zu beliebigen Zeiten gestartet und gestoppt werden können, muss hierfür ein anderes Konzept angewendet werden, das die notwendigen Einträge dynamisch setzt und wieder entfernt, ohne dabei die sonstigen Abläufe zu stören.

Das erledigen die speziell hierfür entworfenen Programme `radacct-start.pl` und `radacct-stop.pl`, die in der `acct_users` festgelegt wurden. Diese Programme müssen mit *sudo* (Superuser)-Rechten ausgeführt werden, weil Änderungen an der Firewall nur mit diesen Rechten möglich sind.

Die beiden Programme bekommen die *Acct-Unique-Session-Id* als Parameter übergeben, die einen eindeutigen Sitzungsidentifikator beinhaltet. Durch eine Datenbankabfrage anhand dieses Wertes ist es den Programmen möglich zu unterscheiden, von welchem Gerätetyp (*nastype*) die Anfrage gestellt wurde. Dies ist für den weiteren Ablauf des Programmes entscheidend, da sowohl für die Access Points als auch für den VPN-Concentrator verschiedene Funktionen definiert sind.

Die nachfolgende Abbildungen zeigen den schematischen Ablauf der Programme für die verschiedenen Gerätetypen:

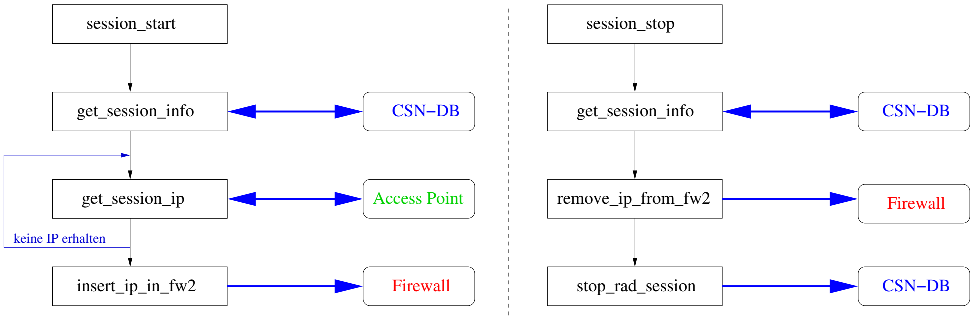

Abbildung 7.3: Schematischer Programmablauf für WLAN mit 802.11i

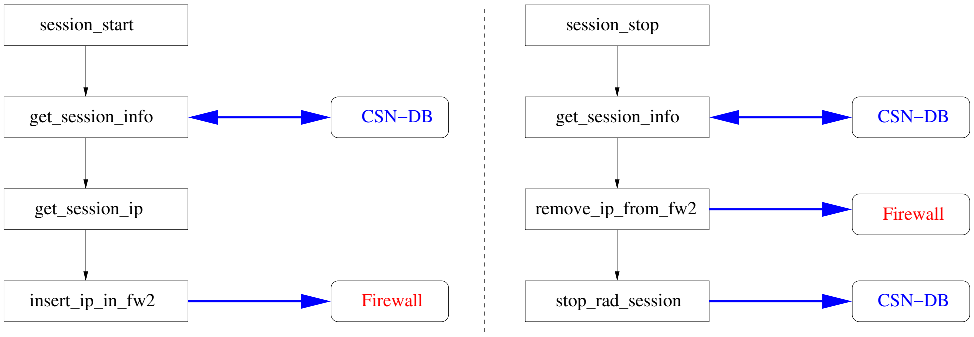

Abbildung 7.4: Schematischer Programmablauf für WLAN mit VPN

7.4.6 FreeRADIUS-Accounting-Erweiterung für CSN-Firewall

Die entwickelte Erweiterung für das FreeRADIUS-Accounting ermittelt für die Sitzung die verwendete IP-Adresse und die unikate NutzerID. Mit diesen Daten werden passende Regeln zum Sitzungsstart in die Firewall eingetragen und zum Sitzungsende wieder entfernt.

Eine nutzerklassenspezifische Einordnung ist nicht vorgesehen, weil nur eine Nutzerklasse das WLAN/VPN nutzen kann. Die zusätzliche Einordnung in Nutzerklassen kann einfach realisiert werden, indem zur NutzerID auch noch die Nutzerklasse aus der Datenbank ausgelesen und der Firewall-Funktion übergeben wird. Diese setzt die Markierung dann entsprechend der Bildungsvorschrift aus Nutzerklasse und NutzerID.

7.4.7 Abrechnung des Datenvolumens

Wie im Abschnitt 7.1.4 bereits beschrieben, werden jede Viertelstunde mittels selbstgeschriebener Programme die kategorisierten Trafficwerte in der Datenbank gespeichert. In der Datenbanktabelle für diese Werte werden nicht nur die IP-Adresse und die Trafficart festgehalten, sondern auch noch die unikate NutzerID. Für alle statisch vergebenen IP-Adressen existiert in weiteren Tabellen eine Relation zwischen IP-Adresse und NutzerID. Ein Datenbank-„Trigger” startet dabei eine Datenbank-Funktion, die beim Einfügen der Trafficwerte die zusätzliche Speicherung der NutzerID in der Traffictabelle erledigt.

Da die IP-Adressen im WLAN/VPN aber dynamisch vergeben werden, gibt es nicht wie üblich eine feste Zuordnung zu einer NutzerID. Um dies zu realisieren, wurde die oben beschriebene Datenbank-Funktion so erweitert, dass sie zusätzlich in der *Accounting*-Tabelle den Nutzer der IP-Adresse sucht und damit die NutzerID ermittelbar und eintragbar ist. Somit wird in der Traffictabelle der Nutzer der IP-Adresse festgehalten. Damit ist das Datenaufkommen der Sitzung dem Nutzer zuordnen und somit abrechenbar.

Die Tabellendefinition mit „Trigger” und die zugehörige Funktion sind im Anhang 9.4.3 zu finden.

Um die Datenbank-Funktion möglichst einfach und stabil zu gestalten, wurden die Werte von *acctstarttime* und *acctstoptime* in der *Accounting*-Tabelle auf volle Viertelstunden gerundet. Warum das möglich ist, ohne dass es dabei zu Überschneidungen kommt, wird nachfolgend kurz erläutert.

WLAN mit 802.11i verwendet einen DHCP-Server für die dynamische IP-Adress-Vergabe. Die Einstellung dessen verhindert, dass die gleiche IP-Adresse im Zeitraum von einem Tag an mehrere unterschiedliche WLAN-Client vergeben werden. Eine Überschneidung in der abzurechnenden Viertelstunde ist deshalb nicht zu erwarten.

VPN-Concentrator hat eine eigene IP-Adress-Vergabe und vergibt jeweils die erste freie IP-Adresse aus dem ihm zugewiesenen IP-Adressraum. Die Vergabe der IP-Adressen kann im System so umgestellt werden, dass die, durch die Trennung eines VPN-Clienten frei gewordene IP-Adresse, für eine gewisse Zeit nicht verwendet werden soll. Dies wurde in diesem Fall auf 30 Minuten gestellt, somit tritt bei der viertelstündigen Abrechnung ebenfalls keine Überschneidung auf.

7.4.8 Netzwerksicherheit

Im kabelgebundenen Netzwerk wird die Sicherheit gegen unbefugte Netzwerkzugriffe mittels MAC-Adress-Filter am jeweiligen Geräte-Port realisiert. Dies ist im WLAN-Netzwerk nicht möglich, weil es keine explizite Anmeldung mit Angabe der MAC-Adresse gibt. Um zu verhindern, dass ein Nutzer sich nach der Anmeldung eine statische IP-Adresse zuweist, um z. B. die Datenvolumenerfassung zu umgehen, wurde auf eine nicht standardisierte Lösung von Cisco Systems zurückgegriffen.

Dabei handelt es sich um eine kombinierte ISO-OSI Layer 2 und 3 Access Control List. Diese wurde an den Geräte-Ports aktiviert, an denen die Access-Points angeschlossen sind. Die Netzkomponente (siehe Anhang 9.3.2.3) registiert die DHCP-Request/Response des WLAN-Clients. Dabei werden die MAC-Adresse und die vom DHCP-Server zugewiesene IP-Adresse sowie die Dauer der Gültigkeit für den jeweiligen WLAN-Client erfasst und gespeichert. Damit legt er automatisch für diesen Port eine ACL fest, die nur Daten mit der entsprechenden Kombination aus MAC- und IP-Adresse im Gültigkeitszeitraum des DHCP-Leases passieren lässt. Eine Änderung oder statische Konfiguration der IP-Adresse ist damit auszuschließen.

Cisco Systems bezeichnet diese beiden Verfahren als *DHCP Snooping* und *IP Source Guard*. Der interessierte Leser kann hierzu ausführliche Informationen auf der Webseite von Cisco Systems [81] nachlesen.

Diese Methode der Netzsicherung spielt nur bei dem WLAN mit 802.11i eine Rolle, da beim VPN sowieso der VPN-Concentrator die IP-Vergabe regelt und somit vom VPN-Client aus nicht umstellbar ist.

Falls ein Access Point in das CSN-Netzwerk integriert werden soll, an dem noch kein aktuelles Gerät mit diesen beiden Technologien vorhanden ist, muss auf das WLAN mit 802.11i aus Sicherheitsgründen verzichtet werden. Ein entsprechender Hinweis sollte bei den Informationen zum WLAN im CSN zu finden sein.

7.4.9 Mögliche Fehlerquellen

Die möglichen Fehlerquellen entstehen hauptsächlich durch das RADIUS-Protokoll an sich, da es sich beim RADIUS zum einen um ein Client/Server-Protokoll handelt und zum anderen einen zustandslosen Automaten darstellt. So kann der Server weder eine Trennung der Station vom NAS veranlassen, noch erwartet er in gewissen Zeitabständen eine Meldung über die noch offenen Sitzungen vom NAS.

So kann es z. B. durch eine Netzunterbrechnung zwischen NAS und RADIUS-Server oder durch einen Systemabsturz zum Verlust des *Accounting-Stop-Request* kommen. Damit gibt es noch offene Sitzungen im RADIUS-Accounting, obwohl die Station nicht mehr am NAS angemeldet ist.

Um solche Fälle zu vermeiden, wurde ein Session-Check-Programm `radacct_session_check.pl` implementiert, das kontinuierlich überprüft, ob die offenen Sitzungen noch auf dem entsprechenden NAS vorhanden sind. Falls nicht, führt dies einen *Accounting-Stop* für die nicht mehr existierende Session aus, das die *acctstoptime* setzt und die Regeln aus der CSN-Firewall entfernt.

Eine Anmeldung am WLAN/VPN während der Netzunterbrechnung zwischen NAS und RADIUS-Server ist nicht möglich, da die NAS-Geräte die Authentifizierungsanfragen nicht weiterleiten können. Es muss also nicht extra beachtet werden.

Um Fehler zu verringern, wurden weiterhin folgende Programme implementiert: `wlan_startup.pl`, `wlan_shutdown.pl` und `wlan_check.pl`

Diese Programme können entweder ohne Parameter oder mit der IP-Adresse eines Access Points aufgerufen werden. Falls sie ohne Parameter aufgerufen werden, wendet das Programm die entsprechenden Aktionen auf alle in der Datenbank als WLAN-Geräte markierten Netzkomponenten an. Falls die IP-Adresse angegeben wird, wird nur das entsprechende Gerät angesprochen. Dabei schaltet `wlan_shutdown.pl` die WLAN-Interfaces der Komponente ab, `wlan_startup.pl` aktiviert sie. `wlan_check.pl` gibt nur den Status des WLAN-Interfaces an.

Falls es zu Wartungsarbeiten an CSN-Servern kommt, sollten zuvor die WLAN-Interfaces deaktiviert werden. Somit werden alle Sitzungen beendet und es sollte zu keinen Fehlern im RADIUS-*Accounting* kommen.

8 Fazit und Perspektiven

8.1 Bewertung

Mit der vorliegenden Software und der zugrundeliegenden Hardware wurde ein mobiler Zugang zum Netzwerk des CSN geschaffen. Dabei wurden zwei Zugangsmöglichkeiten eingerichtet, um eine größtmögliche Anzahl an verschiedenen Betriebssystemen zu unterstützen.

Da für beide Zugänge ein gewisser Konfigurationsaufwand entsteht und eventuell zusätzliche Software installiert werden muss, wurde auf eine gute und einfache Dokumentation Wert gelegt.

Wie sich aus der Erfahrung mit Nutzern herausstellte, wird sehr wenig auf eine angemessene Verschlüsselung der Daten geachtet. Deshalb wurde hier von Grund auf eine Datenverschlüsselung auf der Sicherungs- bzw. Vermittlungs- und Transportschicht des ISO-OSI-Modells gesetzt. Damit werden auch Daten von Anwendung ohne Verschlüsselung geschützt übertragen.

Bei der Implementierung wurde viel Wert auf eine einfache und gut strukturierte Programmierung gelegt, um Erweiterungen oder Änderungen durch die Mitarbeiter des CSN zu erleichtern. Dabei wurde auf die bereits existierenden Softwarebibliotheken und -APIs (Application Program Interfaces) des CSN gesetzt.

8.2 Entwicklungsmöglichkeit

Da auf standardisierte Verfahren aufgesetzt wird, lassen sich diese auch leicht auf andere Netzkomponenten übertragen. So ist es z. B. mit relativ geringem Aufwand möglich, das Anmeldeverfahren vom WLAN mit 802.11i auf Switches mit portbasierender Authentifikation mittels 802.1x anzuwenden. Dazu müssen die Start- und Stop-Programme des RADIUS-*Accountings* erweitert werden.

Außerdem kann durch Änderungen in den Programmen und den Datenbank-„Views” der nutzerklassenbasierende Anschluss auch auf das WLAN angewendet werden.

Durch die Proxy-Fähigkeit des RADIUS-Servers ist auch ein Roaming mit dem RADIUS-Server der TU Chemnitz, dem des DFN (Deutsches Forschungsnetz) oder denen anderer Universitäten denkbar. Damit könnten sich auch Personen, die nicht Nutzer des CSN sind, am WLAN anmelden.

Falls das DIAMETER-Protokoll breiter unterstützt wird, kann dies anstelle des RADIUS eingesetzt werden. Ein Austausch ist dabei relativ einfach möglich. Hierbei muss der FreeRADIUS gegen einen DIAMETER-Server getauscht werden und die NAS-Geräte umkonfiguriert werden.

Abkürzungen

3DES Triple DES

AAA Authentication, Authorization, Accounting

ACL Access Control List

AES Advanced Encryption Standard

AH Authentication Header

API Application Program Interface

ARP Adress Resolution Protocol

ASF Alerting Standards Forum

ASM Application Specific Module

ATM Asynchronous Transfer Mode

AVP Attribute Value Pairs

BNC Bayonet Nut Connector

BPSK Binary Phase Shift Keying

BRAN Broadband Radio Access Network

BSS Basic Service Set

bzw. beziehungsweise

CA Certificate Authority

CBC-MAC Cipher Block Chaining - Message Authentication Code

CCK Complementary Code Keying

CCM CTR/CBC-MAC

CCMP Counter Mode with Cipher Block Chaining Message Authentication Code Protocol

CHAP Challenge Handshake Authentication Protocol

CRC Cyclic Redundancy Check

CSMA/CD Carrier Sense Multiple Access With Collision Detection

CSN Chemnitzer StudentenNetz

CTR Counter

d. h. das heißt

DES Data Encryption Standard

DFS Dynamic Frequency Selection

DHCP Dynamic Host Configuration Protocol

DNS Domain Name Service

DoS Denial-of-Service

DSSS Direct Sequence Spread Spectrum

dt. deutsch

EAP Extensible Authentication Protocol

EAPOL Extensible Authentication Protocol over LAN

EAPOW Extensible Authentication Protocol over WLAN

EIRP Effective Isotropic Radiated Power

EMV Elektromagnetische Verträglichkeit

engl. englisch

ESP Encapsulating Security Payload

ETSI European Telecommunication Standards Institute

ESS Extended Service Set

FCS Frame Check Sequence

FHSS Frequency Hopping Spread Spectrum

GTC Generic Token Card

HiFi High Fidelity

HIPERLAN High Performance Radio Local Area Network

HMAC Hash Message Authentication Code

HomeRF Home Radio Frequency

HTTP HyperText Transfer Protocol

IBSS Independent Basic Service Set

ICV Integrity Check Value

ID Identity

IEEE Institute of Electrical and Electronics Engineers

IETF Internet Engineering Task Force

IKE Internet Key Exchange

IM Instant Messaging

IMAP Internet Message Access Protocol

IP Internet Protocol

IPX Internet Packet Exchange

IR Infrared

ISM Industrial, Scientific, Medical

ISO International Standards Organization

ISM Industrial, Scientific, and Medical Band

IV Initialisierungsvektor

L2F Layer Two Forwarding

L2TP Layer 2 Tunneling Protocol

LAN Local Area Network

LEAP Light Extensible Authentication Protocol (Cisco)

LED Light Emitting Diode

LLC Logical Link Control

LWL Lichtwellenleiter

MD5 Message Digest Algorithm 5

MAC Message Authentication Code / Media Access Control

MIC Message Integrity Check

MMF Multi Mode Fiber - Multi-Mode-Glasfaser

MS-CHAP MicroSoft Challenge Handshake Authentication Protocol

NAK New Authentication Method

NIC Network Interface Card

NIST National Institute of Standards and Technology

OFDM Orthogonality Frequency Division Multiplexing

OTP One Time Password

OSI Open Systems Interconnection

PAE Port Authentication Entities

PAP PPP Authentication Protocol

PC Personal Computer

PDA Personal Digital Assistant

PDU Protocol Data Unit

PEAP Protected Extensible Authentication Protocol

PKI Public Key Infrastruktur

PMK Pairwise Master Key

PN Pseudo Noise

POP3 Post Office Protocol Version 3

PPP Point-to-Point Protocol

PPTP Point-to-Point Tunneling Protocol

PSK Pre-Shared Key

QAM Quadrature Amplitude Modulation

QoS Quality of Service

QPSK Quadrature Phase Shift Keying

RFC Requests for Comments

RSN Robust Security Network

SA Security Association

SIP Session Initiation Protokoll

SMB Server Message Block

SMTP Simple Mail Transfer Protocol

SPI Security Profile Index

SSID Service Set Identity

SSL Secure Socket Layer

TACACS Terminal Access Controller Access Control System

TCP Transmission Control Protocol

TKIP Temporal Key Integrity Protocol

TLS Transport Layer Security

TP Twisted Pair

TPC Transmitter Power Control

TTLS Tunneled Transport Layer Security

TU Technische Universität

u. a. unter anderem

UDP User Datagram Protocol

UNII Unlicensed National Information Infrastructure

URI Uniform Resource Identifier

URL Uniform Resource Locator

URZ Universitätsrechenzentrum

usw. und so weiter

VLAN Virtual LAN

VoIP Voice over IP

VPN Virtual Private Network

WAN Wide Area Network

WEP Wired Equivelent Privacy

WLAN Wireless Local Area Network

WMN Wireless Mesh Network

WPA Wi-Fi Protected Access

WPAN Wireless Personal Area Network

WPP Wireless Performance Prediction

z. B. zum Beispiel

XOR eXclusive OR

ZIN Zertifikat Internet-Nutzung

XTACACS eXtended Terminal Access Controller Access Control System

9 Anhang

9.1 EAP

Paket Typ	Name	Beschreibung
0000 0000	EAP-Packet	Enthält ein EAP-Paket.
0000 0001	EAPoL-Start	Das Benutzersystem kann damit dem Authentifikator den Authentifizierungswunsch signalisieren, der dann mit dem EAP-Request/Identity die Authentifizierung initialisiert.
0000 0010	EAPoL-Logoff	Dient dem Benutzersystem zur Abmeldung und bringt somit den Port in den *unauthorized*-Zustand .
0000 0011	EAPoL-Key	Zum Austausch eines kryptographischen Schlüssels.
0000 0100	EAPoL-Encapsulated - ASF-Alert	Das *Alerting Standards Forum* (ASF) hat einen Weg definiert, um Meldungen mit Hilfe dieses Pakets auch an Ports im *unauthorized* Zustand senden zu können.

Tabelle 9.1: EAP-Paket-Typen

9.2 AAA

9.2.1 RADIUS

Code	Pakettyp
1	Access Request
2	Access Accept
3	Access Reject
4	Accounting Request
5	Accounting Response
11	Access Challenge
12	Status Server (experimental)
13	Status Client (experimental)
255	Reserved

Tabelle 9.2: RADIUS-Codes

Typ	Verwendung
1-63	RADIUS + RADIUS Accounting
64-191	für weitere Spezifizierungen
192-223	für experimentelle Verwendung
224-240	für implementierungsabhängige Verwendung
241-255	vorbehaltlich reserviert

Tabelle 9.3: RADIUS-Attribut-Typen

Typ	Name	Beschreibung
1	User-Name	Benutzername für Anmeldungen
2	User-Password	verschlüsseltes Passwort für PAP
3	CHAP-Password	Antwort auf Challenge bei CHAP
19	Callback-Number	hinterlegte Rufnummer für Rückruf
26	Vendor-Specific	herstellerspezifische Erweiterung
40	Acct-Status-Type	Accounting Start/Stop/Update
60	CHAP-Challenge	hier oder im Feld Authenticator
79	EAP-Message	bei Nutzung des Extensible Authentication Protocol

Tabelle 9.4: RADIUS-Attribute

9.2.2 Diameter

Basistyp	zulässige AVP Länge in Bytes	Beschreibung
OctetString	minimal 8 (12 bei gesetztem V-Bit) - muss auf ein Vielfaches von 32 Bit enden	Dient zum Transport von Texten. Falls der String nicht auf 32 Bit endet, muss bis zur nächsten 32 Bit Grenze aufgefüllt werden.
Integer32	12 (bei gesetztem V-Bit 16)	Eine 32 Bit Integer-Zahl mit Vorzeichen.
Integer64	16 (bei gesetztem V-Bit 20)	Eine 64 Bit Integer-Zahl mit Vorzeichen.
Unsigned32	12 (bei gesetztem V-Bit 16)	Eine 32 Bit Integer-Zahl ohne Vorzeichen.
Unsigned64	16 (bei gesetztem V-Bit 20)	Eine 64 Bit Integer-Zahl ohne Vorzeichen.
Float32	12 (bei gesetztem V-Bit 16)	Eine 32 Bit Zahl mit Nachkommastellen.
Float64	16 (bei gesetztem V-Bit 20)	Eine 64 Bit Zahl mit Nachkommastellen.
Grouped	8 (12 bei gesetztem V-Bit) + die Länge aller in der Gruppe vorhandenen AVPs	Ein grouped AVP fasst unterschiedliche AVPs zu einem AVP zusammen

Tabelle 9.5: DIAMETER-AVP-Basisdatentypen

Name des Typs	abgeleitet von	Beschreibung
Address	OctetString	Beinhaltet eine Adresse, wobei die ersten 2 Octet den Adresstyp spezifizieren.
Time	OctetString	Entspricht den ersten vier Bytes des NTP timestamp Formates.
UTF8String	OctetString	UTF-8 Format
DiameterIdentity	OctetString	Enthält die Identität eines Diameter-Knotens als Fully Qualified Domain Name.
DiameterURI	OctetString	Enthält eine URI eines Diameter-Knotens.
Enumerated	Integer32	Der Wert repräsentiert eine Aufzählung. Die entsprechende Liste wird in der Diameter-Applikation definiert.
IPFilterRule	OctetString	Definiert eine IP Filter Regel.
QoSFilterRule	OctetString	Definiert eine QoS Filter Regel.

Tabelle 9.6: DIAMTER: abgeleitete AVP-Datentypen

9.3 Hardware

9.3.1 Rechnerspezifikation

9.3.1.1 CSN-Server

Hostname	csn-server.csn.tu-chemnitz.de
Standort	Reichenhainer Str. 35
Funktion	- Zentraler Verwaltungsserver - Firewallsystem - Dynamisches Bandbreitenmanagement (Shaper)
CPU	Intel Dual-Xeon 2,4 GHz
RAM	2 GB RAM
NIC	2 Intel Ethernet Pro 10/100/1000 MBit
Betriebssystem	Debian Linux 3.1 (Sarge)
Software	Perl 5.8.4, Postgres 7.4.7, Apache 1.3.33, Bind 9, DHCPd3

9.3.1.2 Accounting-Server

Hostname	wins.csn.tu-chemnitz.de
Standort	Vettersstr. 54
Funktion	- Traffic-Accounting - Samba WINS Dienst - IDS System mit Snort
CPU	Intel Dual-Xeon 2,4 GHz
RAM	2 GB RAM
NIC	2 Intel Ethernet Pro 10/100/1000 MBit
Betriebssystem	Debian Linux 3.1 (Sarge)
Software	Perl 5.8.4, flow-tools, snort

9.3.1.3 Kommunikations-Server

Hostname	pest.csn.tu-chemnitz.de
Standort	Vettersstr. 54
Funktion	- Gateway für Quarantäne-Netz - Gateway für Netze mit privatem IP-Adressraum - Transparenter WebProxy mit Filterregeln
CPU	Intel Celeron 2.4 GHz
RAM	512 MB RAM
NIC	Intel Ethernet Pro 100 + Intel Ethernet Pro 1000
Betriebssystem	Debian Linux 3.1 (Sarge)
Software	Perl 5.8.4, Squid 2.5.9, SquidGuard 1.2

9.3.2 Netzwerksepzifikation

9.3.2.1 Zentraler Router

Hostname	mpr-54-r.csn.tu-chemnitz.de
Standort	Vettersstr. 54
Funktion	- Zentraler Router des CSN - Traffic-Accounting mittels Netflow V5
Hersteller	Cisco Systems
Typ	Cisco Catalyst 6506
Ausstattung	- Supervisor Engine 2a - 48 Port 10/100/1000 MBit/s Modul mit PoE - 8 Port GBic Modul mit 6 SX GBics + 2 LX GBics
Software	Cisco IOS 12.1(26)E2

9.3.2.2 VPN-Concentrator

Hostname	vpngw.csn.tu-chemnitz.de
Standort	Vettersstr. 54
Funktion	- VPN Knotenpunkt des CSN
Hersteller	Cisco Systems
Typ	Cisco VPN 3020 Concentrator
Ausstattung	- 1 Netzteil - 1 Hardware Crypto-Modul
Software	VPN 3000 Concentrator Version 4.7.1.Rel-k9

9.3.2.3 Hausswitch V52

Hostname	mpr-52-kr.csn.tu-chemnitz.de
Standort	Vettersstr. 52
Funktion	- Hausswitch V52 - 288 * 10/100 MBit Ports mit PoE - 7 Geräte gestackt mittels Cisco Stacking-Technologie
Hersteller	Cisco Systems
Typ	6 * Cisco Catalyst 3750-48PS 1 * Cisco Catalyst 3750G-12S
Ausstattung	- Power-over-Ethernet an 288 10/100 MBit/s TP Ports
Software	Cisco IOS 12.2(25)SEB IPBase

9.3.2.4 Access Points

Hostname	ap-52-k[r|l].csn.tu-chemnitz.de
Standort	Vettersstr. 52
Funktion	- Access Point
Hersteller	Cisco Systems
Typ	Cisco Aironet 1230 AG (AIR-AP1231G-E-K9)
Ausstattung	- IEEE 802.11b/g
Software	Cisco IOS 12.3(4)JA

9.4 Datenbankerweiterungen

9.4.1 Authentifizierung und Autorisierung

9.4.1.1 nas

```
CREATE OR REPLACE VIEW nas AS
SELECT
        host_id as id,
        ip_adr as nasname,
        hostname as shortname,
        'other' as type,
        communities[1] as community,
        communities[5] as secret,
        typ as description
FROM hub
WHERE communities[5]<>'';
```

Beispiel:

```
  id  |    nasname    | shortname | type  | community | secret  |  description
------+---------------+-----------+-------+-----------+---------+----------------
   10 | 134.109.100.5 | ap1       | other | public    | geheim  | Cisco 1200 AP
```

9.4.1.2 radcheck

```
CREATE OR REPLACE VIEW radcheck AS
SELECT
        cn.person_id as id,
        cn.urz_login as username,
        character varying 'Crypt-Password' as attribute,
        character varying '==' as op, yp.passwd as value
FROM ypmap yp
JOIN csn_nutzer cn ON cn.urz_login=yp.login
JOIN nutzerrechner nr ON cn.person_id=nr.sv
JOIN host_freizuschalten USING(host_id)
WHERE NOT EXISTS
        ( SELECT 1 FROM csn_nutzer_hat_status
          WHERE person_id=cn.person_id AND status_id=3
          AND CURRENT_TIMESTAMP BETWEEN gueltig_von AND gueltig_bis )
GROUP BY cn.person_id,cn.urz_login,yp.passwd;
```

Beispiel:

```
  id  | username |   attribute    | op |               value
------+----------+----------------+----+------------------------------------
 9787 | aaa      | Crypt-Password | == | $1$e/9aNvOG$xshl8jJcLbssn6VJ9QcGQ0
 9602 | bbb      | Crypt-Password | == | $1$2WA616jh$sSHHWI7gwqgaUkg1rIC63/
```

9.4.1.3 radgroupcheck

```
CREATE OR REPLACE VIEW radgroupcheck AS
        SELECT
                '1' as id,
                'Nutzer' as groupname,
                'Simultaneous-Use' as attribute,
                ':= 'as op,
                '1' as value
        UNION SELECT
                '2' as id,
                'Admin' as groupname,
                'Simultaneous-Use' as attribute,
```

```
                ':= 'as op,
                '10' as value
;
```

Beispiel:

```
 id | groupname |     attribute     | op  | value
----+-----------+------------------+-----+-------
 1  | Nutzer    | Simultaneous-Use | :=  | 1
 2  | Admin     | Simultaneous-Use | :=  | 10
```

9.4.1.4 radusergroup

```
SELECT radcheck.username,
 CASE
  WHEN (EXISTS ( SELECT 1
    FROM csn_aktive_uebersicht2 cau
    WHERE cau.aufgbez::text = 'core-team'::text
    AND radcheck.username::text = cau.urz_login::text))
    THEN 'Admin'::character varying
    ELSE 'Nutzer'::character varying
 END AS groupname
FROM radcheck;
```

Beispiel:

```
 username | groupname
----------+-----------
 aaa      | Nutzer
 bbb      | Admin
```

9.4.1.5 radgroupreply

```
CREATE OR REPLACE VIEW radgroupreply AS
        SELECT
                '1' as id,
                'Nutzer' as groupname,
                'Session-Timeout' as attribute,
                ':= 'as op,
                '86400' as value
        UNION SELECT
                '2' as id,
                'Admin' as groupname,
                'Session-Timeout' as attribute,
                ':= 'as op,
                '' as value
;
```

Beispiel:

```
 id | groupname |    attribute    | op  | value
----+-----------+-----------------+-----+-------
 1  | Nutzer    | Session-Timeout | :=  | 86400
 2  | Admin     | Session-Timeout | :=  |
```

9.4.2 Accounting

9.4.2.1 radacct

```
CREATE TABLE radacct (
```

```
        RadAcctId               BIGSERIAL PRIMARY KEY,
        AcctSessionId           VARCHAR NOT NULL,
        AcctUniqueId            VARCHAR(32) NOT NULL,
        UserName                VARCHAR(32),
        Realm                   VARCHAR(30),
        NASIPAddress            INET NOT NULL,
        NASPortId               INTEGER,
        NASPortType             VARCHAR(32),
        AcctStartTime           TIMESTAMP with time zone,
        AcctStopTime            TIMESTAMP with time zone,
        AcctSessionTime         BIGINT,
        AcctAuthentic           VARCHAR(32),
        ConnectInfo_start       VARCHAR(32),
        ConnectInfo_stop        VARCHAR(32),
        AcctInputOctets         BIGINT,
        AcctOutputOctets        BIGINT,
        CalledStationId         VARCHAR(50),
        CallingStationId        VARCHAR(50),
        AcctTerminateCause      VARCHAR(32),
        ServiceType             VARCHAR(32),
        FramedProtocol          VARCHAR(32),
        FramedIPAddress         INET,
        AcctStartDelay          BIGINT,
        AcctStopDelay           BIGINT
);
CREATE INDEX radacct_active_user_idx ON radacct (userName) WHERE AcctStopTime IS NULL;
CREATE INDEX radacct_start_user_idx ON radacct (acctStartTime, UserName);
```

9.4.2.2 normalize_radius_tf

```
CREATE FUNCTION normalize_radius_tf() RETURNS "trigger"
    AS '
DECLARE
    _hour timestamp;
    _min integer;
    _oid integer;
BEGIN
    IF NEW.acctstarttime > NEW.acctstoptime THEN
        RAISE EXCEPTION ''acctstarttime muss vor acctstoptime liegen!'';
    END IF;
    IF NEW.acctstarttime IS NOT NULL AND NEW.acctstarttime < ''infinity'' THEN
        _hour := date_trunc(''hour'', NEW.acctstarttime);
        _min := trunc(EXTRACT(''minute'' FROM NEW.acctstarttime) / 15) * 15;
        NEW.acctstarttime := _hour + (_min::text || '' minutes'')::interval;
    END IF;
    IF NEW.acctstoptime IS NOT NULL AND NEW.acctstoptime < ''infinity'' THEN
        _hour := date_trunc(''hour'', NEW.acctstoptime);
        _min := trunc(EXTRACT(''minute'' FROM NEW.acctstoptime) / 15) * 15;
        IF _hour + (_min::text || ''minutes'')::interval < NEW.acctstoptime THEN
            NEW.acctstoptime := _hour + ((_min+15)::text || '' minutes'')::interval;
        END IF;
    END IF;
    IF TG_OP = ''INSERT'' THEN
        SELECT oid INTO _oid FROM radacct
            WHERE username = NEW.username AND framedipaddress = NEW.framedipaddress
            AND COALESCE(acctstoptime, ''infinity'') >= NEW.acctstarttime
            AND COALESCE(acctstoptime, ''infinity'') < COALESCE(NEW.acctstoptime, ''infinity'');
        IF FOUND THEN
            UPDATE radacct SET acctstoptime = NEW.acctstoptime WHERE oid = _oid;
            RETURN NULL;
        END IF;
    END IF;
    RETURN NEW;
END;
' LANGUAGE plpgsql;

CREATE TRIGGER normalize_radius_t
    BEFORE INSERT OR UPDATE ON radacct
    FOR EACH ROW
    EXECUTE PROCEDURE normalize_radius_tf();
```

9.4.3 Trafficaccounting

9.4.3.1 traffic_current

```
CREATE TABLE traffic_current (
        ip_adr          inet NOT NULL,
        host_id         integer REFERENCES nutzerrechner(host_id) DEFAULT NULL,
        person_id       integer REFERENCES csn_nutzer(person_id) DEFAULT NULL,
        von             timestamp NOT NULL,
        bis             timestamp NOT NULL,
        trafficart_id   integer NOT NULL REFERENCES trafficart(trafficart_id),
        traffic_up      bigint NOT NULL,
        traffic_down    bigint NOT NULL
);
CREATE INDEX traffic_current_host_id ON traffic_current(host_id);
CREATE INDEX traffic_current_person_id ON traffic_current(person_id);
CREATE INDEX traffic_current_bis ON traffic_current(bis);
```

9.4.3.2 traffic_current_person_id_tf

```
CREATE OR REPLACE FUNCTION traffic_current_person_id_tf() RETURNS TRIGGER AS '
DECLARE
        _person_id integer;
        _host_id integer;
BEGIN
        IF NEW.person_id IS NOT NULL THEN
                RETURN NEW;
        END IF;
        SELECT sv INTO _person_id FROM nutzerrechner WHERE ip_adr = NEW.ip_adr;
        SELECT host_id INTO _host_id FROM nutzerrechner WHERE ip_adr = NEW.ip_adr;
        IF FOUND THEN
                NEW.person_id = _person_id;
                NEW.host_id = _host_id;
                RETURN NEW;
        END IF;
        SELECT person_id INTO _person_id
          FROM radacct JOIN csn_nutzer ON username = urz_login
          WHERE framedipaddress = NEW.ip_adr
          AND NEW.von + ''10 seconds''::interval BETWEEN acctstarttime AND COALESCE(acctstoptime, ''infinity'');
        IF FOUND THEN
                NEW.person_id = _person_id;
                RETURN NEW;
        END IF;
        RETURN NEW;
END;
' LANGUAGE 'plpgsql';

CREATE TRIGGER traffic_current_person_id_t
  BEFORE INSERT OR UPDATE ON traffic_current
  FOR EACH ROW EXECUTE PROCEDURE traffic_current_person_id_tf();
```

9.5 Konfigurationen

9.5.1 FreeRadius

9.5.1.1 radiusd.conf

```
...
$INCLUDE ${confdir}/eap.conf
$INCLUDE  ${confdir}/postgresql.conf
...
files {
        ...
        acctusersfile = ${confdir}/acct_users
}
...
```

```
authorize {
           eap
           sql
           ...
}
authenticate {
        Auth-Type PAP { pap }
        ...
        eap
}
preacct {
        ...
        #  Read the 'acct_users' file
        files
}
accounting {
            sql
}
session {
         sql
}
...
```

9.5.1.2 postgresql.conf

```
sql {
        driver = "rlm_sql_postgresql"
        server = ""
        port="5430"
        login = "freerad"
        password = "****"
        radius_db = "csn"
        acct_table1 = "radacct"
        acct_table2 = "radacct"
        authcheck_table = "radcheck"
        groupcheck_table = "radgroupcheck"
        groupreply_table = "radgroupreply"
        usergroup_table = "radusergroup"
        nas_table="nas"
        ...
        readclients=yes
}
```

9.5.1.3 eap.conf

```
eap {
    ...
    tls {
          private_key_file = ${raddbdir}/certs/server.key
          certificate_file = ${raddbdir}/certs/server.crt
          CA_file = ${raddbdir}/certs/ca-urz.crt
          dh_file = ${raddbdir}/certs/dh
          random_file = ${raddbdir}/certs/random
          fragment_size = 1024
          include_length = yes
     }
     ttls {
           default_eap_type = md5
           copy_request_to_tunnel = no
           use_tunneled_reply = no
          }
     ...
}
```

9.5.1.4 acct_users

```
DEFAULT Acct-Status-Type == Start
       Exec-Program = "/usr/bin/sudo /var/CSN/bin/radacct-start.pl %{Acct-Unique-Session-Id}"

DEFAULT Acct-Status-Type == Stop
       Exec-Program = "/usr/bin/sudo /var/CSN/bin/radacct-stop.pl %{Acct-Unique-Session-Id}"
```

9.5.2 Zugangskontroll-Server

Software für die Umsetzung des transparenten Proxy:

- IPTables (Netfilter [68])
- Squid [82]
- squidGuard [83]

9.5.2.1 IPTables-Konfiguration

```
...

# DNS erlauben
iptables -A FORWARD -d 134.109.102.249 -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -s 134.109.102.249 -p udp --sport 53 -j ACCEPT
iptables -A FORWARD -d 134.109.102.1 -p udp --dport 53 -j ACCEPT
iptables -A FORWARD -s 134.109.102.1 -p udp --sport 53 -j ACCEPT
# wlan_netz einhaengen
iptables -A FORWARD -o eth0 -s 172.16.10.0/23 -j wlan_netz
iptables -A FORWARD -i eth0 -d 172.16.10.0/23 -j wlan_netz
iptables -N wlan_netz
iptables -F wlan_netz
iptables -A wlan_netz -s 172.16.10.0/23 -d 134.109.102.249 -p tcp --dport 443 -j ACCEPT
iptables -A wlan_netz -s 134.109.102.249 -d 172.16.10.0/23 -p tcp --sport 443 -j ACCEPT
iptables -A wlan_netz -s 172.16.10.0/23 -d 134.109.102.249 -p tcp --dport 80 -j ACCEPT
iptables -A wlan_netz -s 134.109.102.249 -d 172.16.10.0/23 -p tcp --sport 80 -j ACCEPT
iptables -A wlan_netz -j DROP

...
# redirect fuer squid
iptables -t nat -A PREROUTING ! -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 3128
iptables -t nat -A PREROUTING ! -i eth0 -p tcp --dport 8080 -j REDIRECT --to-port 3128
```

9.5.2.2 squid.conf

```
...
redirect_program /usr/bin/squidGuard -c /etc/squid/squidGuard.conf
...
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
....
```

9.5.2.3 squidGuard.conf

```
...
src wlan_clients {
        ip 172.16.10.0/23
        logfile wlan
}
...
dest csn_domain {
        domainlist /etc/squid/csn_domain
        logfile csn_log
}
...
acl {
        wlan_clients {
                pass csn_domain none
                redirect 302:http://www.csn.tu-chemnitz.de/info/wlan/index.html
        }
```

```
...
}
```

9.5.3 Access Points

9.5.3.1 WLAN mit 802.11i

Teil der Konfiguration des Access Points (siehe Anhang 9.3.2.4), der für das WLAN mit 802.11i wichtig ist:

```
aaa group server radius rad_eap
 server 134.109.102.249 auth-port 1812 acct-port 1813
!
aaa group server radius rad_acct
 server 134.109.102.249 auth-port 1812 acct-port 1813
!
aaa authentication login eap_methods group rad_eap
aaa accounting network acct_methods start-stop group rad_acct
aaa session-id common
!
dot11 ssid CSN_WPA
   vlan 90
   authentication open eap eap_methods
   authentication network-eap eap_methods
   authentication key-management wpa
   accounting acct_methods
   mbssid guest-mode
!
interface Dot11Radio0
 encryption vlan 90 mode ciphers aes-ccm tkip
 broadcast-key vlan 90 change 3600
!
radius-server attribute 32 include-in-access-req format %h
radius-server host 134.109.102.249 auth-port 1812 acct-port 1813 key 0 geheimwort
radius-server vsa send accounting
!
```

9.5.3.2 WLAN mit 802.11i

Teil der Konfiguration des Access Points (siehe Anhang 9.3.2.4), der für das offene WLAN wichtig ist:

```
dot11 ssid CSN
   vlan 10
   authentication open
   mbssid guest-mode
!
interface Dot11Radio0
 ssid CSN
!
```

9.5.3.3 Ethernet-Port

Teil der Konfiguration des Access Points (siehe Anhang 9.3.2.4), der für das Ethernet-Interface wichtig ist:

```
interface FastEthernet0.1
 encapsulation dot1Q 1 native
 no ip route-cache
 bridge-group 1
 no bridge-group 1 source-learning
 bridge-group 1 spanning-disabled
```

```
!
interface FastEthernet0.10
 encapsulation dot1Q 10
 no ip route-cache
 bridge-group 10
 no bridge-group 10 source-learning
 bridge-group 10 spanning-disabled
!
interface FastEthernet0.90
 encapsulation dot1Q 90
 no ip route-cache
 bridge-group 90
 no bridge-group 90 source-learning
 bridge-group 90 spanning-disabled
!
```

9.5.4 Switch

9.5.4.1 Switchports für die Access Points

Teil der Konfiguration des Switches (siehe Anhang 9.3.2.3), der für die Access Points wichtig ist:

```
ip dhcp snooping vlan 10,90
no ip dhcp snooping information option
ip dhcp snooping database flash:dhcp-snooping
ip dhcp snooping
!
interface FastEthernet3/0/43
 description AP1
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 103
 switchport trunk allowed vlan 10,90,103
 switchport mode trunk
 ip verify source port-security
!
interface FastEthernet3/0/44
 description AP2
 switchport trunk encapsulation dot1q
 switchport trunk native vlan 103
 switchport trunk allowed vlan 10,90,103
 switchport mode trunk
 ip verify source port-security
!
```

Beispiel des DHCP Snooping und IP Source Guard:

```
MacAddress          IpAddress        Lease(sec)  Type           VLAN  Interface
------------------  ---------------  ----------  -------------  ----  --------------------
00:0C:23:45:EE:00   134.109.90.1     679         dhcp-snooping  90    FastEthernet3/0/43
00:E0:77:AA:CC:11   172.16.10.25     640         dhcp-snooping  10    FastEthernet3/0/44
```

Literaturverzeichnis

[1] URZ - Universitätsrechenzentrum. http://www.tu-chemnitz.de/urz/.

[2] ZIN - Zertifikat Internet-Nutzung. http://www.tu-chemnitz.de/urz/zin/.

[3] Markus Schade. Netzentwicklung im CSN. http://archiv.tu-chemnitz.de/pub/2004/0135/, 2004.

[4] Jan Horbach. Dynamische Bandbreitenbeschränkung mit QoS. http://archiv.tu-chemnitz.de/pub/2001/0100/, 2001.

[5] BNA - Die Bundesnetzagentur. http://www.bundesnetzagentur.de/.

[6] Jörg Roth. *Mobile Computing*. 2002.

[7] Axel Sikora. *Wireless LAN*. Addision-Wesley Verlag, 2001.

[8] Martin Sauter. *Grundkurs Mobile Kommunikationssysteme*. 2004.

[9] A. Merkle und A. Terzis. *Digitale Funkkommunikation mit Bluetooth*. 2002.

[10] Matthew Gast. *802.11 Wireless Networks - The Definitive Guide*. O'Reilly, 2002.

[11] Bruce Schneider. *Angewandte Kryptographie (5. Auflage)*. 1996.

[12] J. Daemen and V. Rijmen. Rijndael the advanced encryption standard (Dr. Dobbs Journal 26 (3) S. 137-139.), 2001.

[13] National Institute of Standards and Technology. Advanced Encryption Standard (AES). http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf, 2001.

[14] S. Kent and R. Atkinson. RFC-2406 - IP Encapsulating Security Payload (ESP), 1998.

[15] S. Kent and R. Atkinson. RFC-2402 - IP Authentication Header, 1998.

[16] D. Harkins and D. Carrel. RFC-2409 - The Internet Key Exchange (IKE), 1998.

[17] C. Madson and R. Glenn. RFC-2403 - The Use of HMAC-MD5-96 within ESP and AH, 1998.

[18] C. Madson and R. Glenn. RFC-2404 - The Use of HMAC-SHA-1-96 within ESP and AH, 1998.

[19] Jon C. Snader. *VPNs Illustrated: Tunnels, VPNs, and IPsec*. 2005.

[20] Naganand Doraswamy and Dan Harkins. *IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks, Second Edition*. 2003.

[21] W. Simpson. RFC 1661 - The Point-to-Point Protocol (PPP), 1994.

[22] S. Hanks, T. Li, D. Farinacci, and P. Traina. RFC-1701 - Generic Routing Encapsulation (GRE), 1994.

[23] S. Hanks, T. Li, D. Farinacci, and P. Traina. RFC-1702 - Generic Routing Encapsulation over IPv4 networks, 1994.

[24] B. Lloyd and W. Simpson. RFC-1334 - PPP Authentication Protocols (PAP), 1992.

[25] W. Simpson. RFC-1994 - PPP Challenge Handshake Authentication Protocol (CHAP), 1996.

[26] K. Hamzeh, G. Pall, W. Verthein, J. Taarud, W. Little, and G. Zorn. RFC-2637 - Point-to-Point Tunneling Protocol (PPTP), 1999.

[27] W. Townsley, A. Valencia, A. Rubens, G. Pall, G. Zorn, and B. Palter. RFC 2661 - Layer Two Tunneling Protocol (L2TP), 1999.

[28] A. Valencia, M. Littlewood, and T. Kolar. RFC 2341 - Cisco Layer Two Forwarding (Protocol) (L2F), 1998.

[29] B. Patel, B. Aboba, W. Dixon, G. Zorn, and S. Booth. RFC 3193 - Securing L2TP using IPsec, 2001.

[30] T. Dierks and C. Allen. RFC 2246 - The TLS Protocol Version 1.0, 1999.

[31] VTun. http://vtun.sourceforge.net/.

[32] CIPE. http://sites.inka.de/sites/bigred/devel/cipe.html.

[33] OpenVPN. http://openvpn.net/.

[34] L. Blunk and J. Vollbrecht. RFC-2284 - PPP Extensible Authentication Protocol (EAP), 1998.

[35] xsupplicant. http://open1x.sourceforge.net/.

[36] wpa_supplicant. http://hostap.epitest.fi/wpa_supplicant/.

[37] C. Rigney, A. Rubens, W. Simpson, and W. Simpson. RFC-2138 - Remote Authentication Dial In User Service (RADIUS), 1997.

[38] C. Rigney. RFC-2139 - RADIUS Accounting, 1997.

[39] C. Rigney, A. Rubens, W. Simpson, and W. Simpson. RFC-2865 - Remote Authentication Dial In User Service (RADIUS), 2000.

[40] B. Aboba and P. Calhoun. RFC-3579 - RADIUS Support For Extensible Authentication Protocol (EAP), 2003.

[41] C. Rigney, W. Willats, and P. Calhoun. RFC-2869 - RADIUS Extensions, 2000.

[42] P. Congdon, B. Aboba, A. Smith, G. Zorn, and J. Roese. RFC-3580 - IEEE 802.1X Remote Authentication Dial In User Service (RADIUS) Usage Guidelines, 2003.

[43] H. Krawczyk, M. Bellare, and R. Canetti. RFC-2104 - HMAC: Keyed-Hashing for Message Authentication, 1997.

[44] C. Rigney. RADIUS Accounting, 2000.

[45] G. Zorn, D. Leifer, D. Leifer, J. Shriver, M. Holdrege, and I. Goyret. RFC-2868 - RADIUS Attributes for Tunnel Protocol Support, 2000.

[46] Cisco Secure Access Control Server. http://www.cisco.com/go/acs.

[47] FreeRADIUS. http://www.freeradius.org/.

[48] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, and J. Arkko. RFC-3588 - Diameter Base Protocol, 2003.

[49] P. Calhoun, T. Johansson, C. Perkins, T. Hiller, and P. McCann. RFC-4004 - Diameter Mobile IPv4 Application, 2005.

[50] P. Calhoun, G. Zorn, D. Spence, and D. Mitton. RFC-4005 - Diameter Network Access Server Application, 2005.

[51] H. Hakala, L. Mattila, J-P. Koskinen, M. Stura, and J. Loughney. RFC-4006 - Diameter Credit-Control Application, 2005.

[52] P. Eronen, T. Hiller, and G. Zorn. RFC-4072 - Diameter Extensible Authentication Protocol (EAP) Application, 2005.

[53] R. Stewart, Q. Xie, K. Morneault, C. Sharp, H. Schwarzbauer, T. Taylor, I. Rytina, M. Kalla, L. Zhang, and V. Paxson. RFC-2960 - Stream Control Transmission Protocol, 2000.

[54] OpenDiameter. http://www.opendiameter.org/.

[55] HP-UX Mobile AAA Server. http://www.hp.com/.

[56] DIameter Server Client. http://developer.berlios.de/projects/disc/.

[57] C. Finseth. RFC-1492 - An Access Control Protocol, Sometimes Called TACACS, 1993.

[58] TACACS. http://ing.ctit.utwente.nl/WU5/D5.1/Technology/tacacs/.

[59] Radtac Manager Server. http://www.radtac.com/.

[60] HandyTacacs 1.0. http://www.arithmetica.ch/HandyTacacs/.

[61] XTACACS. http://ing.ctit.utwente.nl/WU5/D5.1/Technology/xtacacs/.

[62] David Carrel and Lol Grant. The TACACS+ Protocol Version 1.78. ftp://www.ietf.org/internet-drafts/draft-grant-tacacs-02.txt, 1998.

[63] TACACS+. http://ing.ctit.utwente.nl/WU5/D5.1/Technology/tacacs+/.

[64] Mirko Parthey. Zugangsmanagement für Wireless LAN. http://archiv.tu-chemnitz.de/pub/2002/0106/, 2001.

[65] Gonzalo Camarillo and Miguel A. Garcia-Martin. *The 3G IP Multimedia Subsystem*. Wiley, 2004.

[66] Jeffrey Bannister, Paul Mather, and Sebastian Coope. *Convergence Technologies for 3G Networks*. Wiley, 2004.

[67] Jabber. http://www.jabber.org/.

[68] Netfilter - Firewalling, NAT and packet mangling for Linux. http://www.netfilter.org.

[69] Sebastian Junge. Klassenbasierter Anschluss im Chemnitzer Studenten Netz. http://archiv.tu-chemnitz.de/pub/2005/0168/, 2005.

[70] Markus Schade and Daniel Schreiber. A Packet filtering Concept based on iptables MARK's. https://www.csn.tu-chemnitz.de/~marks/firewall.pdf, 2005.

[71] Cisco IOS NetFlow. http://www.cisco.com/go/netflow.

[72] Flow-Tools. http://www.splintered.net/sw/flow-tools/.

[73] Virtual Bridged LANs. http://standards.ieee.org/getieee802/download/802.1Q-2003.pdf, 2003.

[74] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, and E. Lear. RFC-1918 - Address Allocation for Private Internets, 1996.

[75] CSN Mobil. https://www.csn.tu-chemnitz.de/info/wlan/index.html.

[76] J. Franks, J. Franks, J. Hostetler, S. Lawrence, P. Leach, P. Leach, and P. Leach. RFC-2617 - HTTP Authentication: Basic and Digest Access Authentication, 1999.

[77] Debian. http://www.debian.org.

[78] SecureW2. http://www.securew2.com/.

[79] Cisco VPN 3000 Series Concentrators. http://www.cisco.com/go/vpn3k.

[80] vpnc - client for cisco3000 VPN Concentrator. http://www.unix-ag.uni-kl.de/~massar/vpnc/.

[81] Cisco Systems. http://www.cisco.com.

[82] Squid Web Proxy Cache. http://www.squid-cache.org/.

[83] squidGuard - An ultrafast and free filter/redirector/access controller for Squid. http://www.squidguard.org/.

Abbildungsverzeichnis

Tabellenverzeichnis

Erklärungen

Haftungsausschluss

Der Autor, das Chemnitzer StudentenNetz und die Technische Universitat Chemnitz übernehmen keine Haftung für Schäden, die sich aus der Benutzung dieser Arbeit oder der zugehörigen Software ergeben.

Schutzrechte

Eingetragene Waren- und Markenzeichen sind im Text nicht als solche gekennzeichnet. Das Fehlen der Kennzeichnung bedeutet nicht, dass diese Zeichen frei verwendbar sind.

Mir sind zum Zeitpunkt der Abgabe keine Ansprüche Dritter an dieser Arbeit und der zugehörigen Software bekannt.

Selbständigkeitserklärung

Ich erkläre hiermit, dass ich die vorliegende Diplomarbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Chemnitz, den 14. Dezember 2005